SecWeb: Privacy-Preserving Web Browsing Monitoring with w-Event Differential Privacy

Qian Wang[1(✉)], Xiao Lu[1], Yan Zhang[1], Zhibo Wang[1], Zhan Qin[2], and Kui Ren[2]

[1] The State Key Lab of Software Engineering, School of CS, Wuhan University, Wuhan, China
{qianwang,luxiao,stong,zbwang}@whu.edu.cn
[2] Department of CSE, The State University of New York, Buffalo, USA
{zhanqin,kuiren}@buffalo.edu

Abstract. Nowadays aggregated web browsing histories of individual users have been collected and extensively used by Internet service providers as well as third-party researchers, due to their great value to data mining for in-depth understanding of important phenomena, such as suspicious behavior detection. While providing tremendous benefits, the release of private users' data to the public will pose a considerable threat to users' privacy. Sharing web browsing data with privacy preservation has so far received very limited research attention. In this paper, we investigate the problem of real-time privacy-preserving web browsing monitoring, and propose SecWeb, an online aggregated web browsing behavior monitoring scheme over infinite time with theoretical privacy guarantee. Specifically, we propose an adaptive sampling mechanism and an adaptive budget allocation mechanism to better allocate appropriate privacy budget to sampling points within any successive w time stamps. In addition, we propose a dynamic grouping mechanism that groups web pages with small visits together and adds Laplace noise to each group instead of single web page to eliminate the effects of perturbation error for the web pages. We prove that SecWeb satisfies w-event differential privacy and the experimental results on a real-world dataset show that SecWeb outperforms the state-of-the-art approaches.

Keywords: Web browsing · Privacy preservation · Real-time data publishing · Differential privacy · w-event privacy

1 Introduction

The Internet plays a more and more important role in people's daily life as the explosive growth of mobile devices. People can obtain their interested information by browsing various websites, while their browsing behaviors which can be characterized by browsing histories are also recorded by the host servers simultaneously. The servers may publish the browsing histories[1] to the public since

[1] In this paper, we interchangeably use "web browsing histories" and "web browsing data" throughout the paper without confusion.

R. Deng et al. (Eds.): SecureComm 2016, LNICST 198, pp. 454–474, 2017.
DOI: 10.1007/978-3-319-59608-2_26

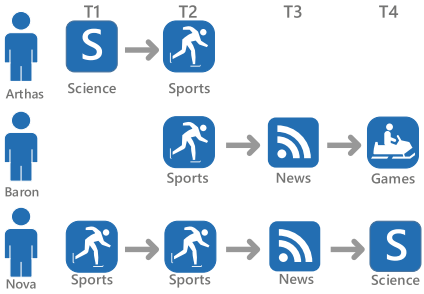


Fig. 1. An example of web browsing behaviors

Table 1. Statistics of web browsing behaviors

	News	Games	Sports	Science
T1	0	0	1	1
T2	0	0	3	0
T3	2	0	0	0
T4	0	1	0	1

these data is of great value for companies or third-party researchers in many data mining applications to analyze the browsing behavior of users. It is also strategic significance for understanding the users' habits in order to improve the user experience and websites performance, such as recommending web pages to users based on their browsing behavior, finding the current hot news, and watching the network traffic to detect anomaly [5].

However, there is always a risk in releasing this kind of private and sensitive data to the public. Studies have indicated that a user's web browsing history (i.e., a sequence of visited websites) can be regarded as a fingerprint which can be used to uniquely identify or track the user [21]. The AOL data release [4] is a representative privacy incident where a newspaper journalist quickly identified a user by the released anonymized search logs and consequently the sensitive information of this user was disclosed. This and other related findings indicate that the released private data must be carefully processed to protect the privacy of individuals [25].

Generally speaking, different people behave different web browsing patterns. Thus, people's sensitive information can be easily figured out by exploiting users' browsing histories. Even the identification information is hidden from the public, it is still possible to discover the browsing histories of users. For example, Fig. 1 illustrates the browsing behaviors of three people, e.g., Baron starts his browsing session at time stamp $T2$, visiting a sports news page, then he moves to a local news page and his session ends after browsing a web page about games. Table 1 shows the number of visits to each type of web page without any identification information. With background information, the adversary knows that Baron

starts to browse the web page at time stamp $T2$. Then the adversary can easily obtain two browsing traces for Baron, i.e., *sports* → *news* → *games* or *sports* → *news* → *science*, from the released data. Suppose the adversary already knows that Baron is interested in playing games from the side channel information, e.g., the public tweet from Twitter. Then he can infer that *sports* → *news* → *games* is more likely to be the true browsing trace of Baron. Therefore, it is important and necessary to process the web browsing data before publishing so that the released data is not only useful but also privacy-preserved.

The technique of differential privacy (DP) [9] can ensure privacy protection for statistic data publishing with vigorous guarantee theoretically. Now it has become an appealing privacy model. In particular, DP does not need to make any assumption about the adversary's background information. That is, even the adversary has obtained a user's background information, it cannot derive any additional information about the user based on his/her published data.

Almost all of the existing differentially private protocols investigated either event-level privacy on infinite streams [6,7,11] or user-level privacy on finite streams [12,13]. The authors in [18] successfully bridged the gap between the user-level and the event-level over streams using the *w-event ϵ-differential privacy* model (i.e., w-event privacy) to make a good trade-off between the privacy and the utility, and thus it can protect any sequence of events existing within any time stamp window of length w.

In this paper, we investigate the real-time web browsing data publishing problem with privacy protection, e.g., securing the number of visits to different web pages at each time stamp. [12] took the first step to share web browsing data with differential privacy, which focused on real-time web browsing data release over a pre-specified finite time stamps. However, the continuous publication of web browsing data (called data streaming) may further reveal sensitive information of users, which motivates the research on privacy preserving real-time web browsing data publishing over infinite time. The w-event privacy model well suits for the infinite stream case, and it can provide a full protection of any user's browsing traces (e.g., a sequence of visited web pages) over any sequence of continuous time stamps of length w. We summarize the main contributions of this paper as follows.

- We design a novel privacy preserving scheme, called SecWeb, for real-time web browsing data publishing with strong privacy guarantee. We design a dynamic grouping mechanism which groups all web pages with a small number of visits, and Laplace noise is inserted to every group other than a single web page to eliminate the effects of perturbation error on web pages.
- We propose adaptive sampling and budget allocation schemes to better allocate appropriate budget of privacy to the sampling points within any sequence of continuous time stamps of length w. We further propose a pre-sampling mechanism to reduce the high query sensitivity and integrate it with SecWeb seamlessly.
- We theoretically prove that SecWeb satisfies the notion of w-event ϵ-differential privacy. SecWeb is evaluated with a real-world dataset,

and compare it with the state-of-the-art approaches. The results demonstrate that SecWeb outperforms the previous approaches and improves the utility or accuracy of real-time web browsing data publishing with a vigorous privacy guarantee.

The remaining parts of this paper are organized as the following sections. Section 2 introduces some preliminary knowledge, describes the problem formulation and briefly discuss the related works. We present SecWeb and analyze its privacy in Sect. 3. We evaluate the performance of SecWeb with extensive experiments in Sect. 4 and finally conclude the paper in Sect. 5.

2 Backgroud

2.1 Preliminaries and Problem Statement

In this section, we introduce some preliminary knowledge of differential privacy and w-event privacy, and present the problem to be studied in this paper.

Differential Privacy. Differential privacy has become a *de-facto* standard privacy model for statistics analysis with provable privacy guarantee. Intuitively, a mechanism satisfies differential privacy if its outputs are approximately unchanged even if a record in the dataset is removed, so that an adversary infers no more information about the record from the mechanism outputs.

Definition 1 (Differential Privacy [9]). *A privacy mechanism $\mathcal{M}$ gives ϵ-differential privacy, where $\epsilon > 0$, if for any datasets D and D' differing on at most one record, and for all sets $S \subseteq Range(\mathcal{M})$,*

$$\Pr[\mathcal{M}(D) \in S] \leq \exp(\epsilon) \cdot \Pr[\mathcal{M}(D') \in S], \tag{1}$$

where ϵ is the *privacy budget* representing the privacy level the mechanism provides. Generally speaking, a smaller ϵ guarantees a stronger privacy level.

Definition 2 (l_1-norm Sensitivity [10]). *For any function $f : \mathcal{D} \rightarrow \mathcal{R}^d$, the l_1-norm sensitivity of f w.r.t. $\mathcal{D}$ is*

$$\Delta(f) = \max_{D, D' \in \mathcal{D}} ||f(D) - f(D')||_1 \tag{2}$$

for all D, D' differing on at most one record.

Laplace mechanism is commonly used to realize ϵ-differential privacy, which adds noise drawn from a Laplace distribution into the datasets to be published.

Theorem 1 (Laplace Mechanism [10]). *For any function $f : \mathcal{D} \rightarrow \mathcal{R}^d$, the Laplace Mechanism $\mathcal{M}$ for any dataset $D \in \mathcal{D}$*

$$\mathcal{M}(D) = f(D) + \langle Lap(\Delta(f)/\epsilon) \rangle^d \tag{3}$$

satisfies ϵ-differential privacy, where the noise $Lap(\Delta(f)/\epsilon)$ is drawn from a Laplace distribution with mean zero and scale $\Delta(f)/\epsilon$.

Intuitively, the noise is large if sensitivity $\Delta(f)$ is big or the budget ϵ is small.

w-Event Privacy. The notion of w-event ϵ-differential privacy (i.e., w-event privacy) was first proposed in [18]. This new privacy model can give provable privacy assurance for any sequence of events within successive time stamps of length w.

Before giving the formal definition of *w-event privacy*, we first introduce some necessary notions. Two data sets D_i, D_i' at time stamp i are neighboring if they have at most one different row. At time stamp t, we define the stream prefix of an infinite series $S = (D_1, D_2, ...)$ as $S_t = (D_1, D_2, ..., D_t)$.

Definition 3 (w-neighboring [18]). *w is a positive integer, we say that S_t, $S_t^{'}$ are w-neighboring, if*

1. *For every $S_t[i]$, $S_t'[i]$ such that $i \in [t]$ and $S_t[i] \neq S_t'[i]$, it holds that $S_t[i]$, $S_t'[i]$ are neighboring, and*
2. *For every $S_t[i_1]$, $S_t[i_2]$, $S_t'[i_1]$, $S_t'[i_2]$ with $i_1 < i_2$, $S_t[i_1] \neq S_t'[i_1]$ and $S_t[i_2] \neq S_t'[i_2]$, it holds that $i_2 - i_1 + 1 \leq w$.*

Simply put, if S_t, S_t' are w-neighboring, their elements are pairwise the same or neighboring, and the time interval of any two neighboring datasets will not exceed w time stamps.

Definition 4 (w-Event Privacy [18]). *A mechanism $\mathcal{M}$ satisfies w-event ϵ-differential privacy, if for all sets $S \subseteq Range(\mathcal{M})$ and all w-neighboring stream prefixes S_t, S_t' and all t, it holds that*

$$\Pr[\mathcal{M}(S_t) \in S] \leq exp(\epsilon) \cdot \Pr[\mathcal{M}(S_t') \in S]. \tag{4}$$

A mechanism satisfying w-event privacy will protect the sensitive information that may be disclosed from a sequence of some length w.

Theorem 2 ([18]). *Let $\mathcal{M}$ be a mechanism that takes stream prefix S_t as input, where $S_t[i] = D_i \in \mathcal{D}$, and outputs $\boldsymbol{s} = (\boldsymbol{s}_1, ..., \boldsymbol{s}_t) \in Range(\mathcal{M})$. Suppose $\mathcal{M}$ can be decomposed into t mechanisms $\mathcal{M}_1, ..., \mathcal{M}_t$ such that $\mathcal{M}_i(D_i) = \boldsymbol{s}_i$, each $\mathcal{M}_i$ generates independent randomness and achieves ϵ_i-differential privacy. Then, $\mathcal{M}$ satisfies w-event privacy if*

$$\forall i \in [t], \sum_{k=i-w+1}^{i} \epsilon_k \leq \epsilon. \tag{5}$$

This theorem enables us to view ϵ as the total available privacy budget in any sliding window of size w, and appropriately allocate portions of it across the time stamps.

Problem Statement. In this paper, we consider the application of continually publishing web browsing histories to the public in real-time manner and aim to realize w-event privacy for real-time web browsing data publishing. Here, a browsing history is defined as a sequence of web pages browsed at consecutive and discrete time stamps. A host server collects and records users' browsing data, and generates a database D as time goes. The objective is to continually make the data statistics calculated on D public in a real-time manner with the guarantee of w-event privacy. To achieve this goal, the host server will not release the real value of statistics, but apply a well-designed privacy protection scheme to publish a sanitised version of the original statistics.

The server gathers the users' browsing logs throughout the time, and at time stamp i obtains a two-dimensional database/matrix D_i, where the columns correspond to the web pages and the rows correspond to the users. $D_i[m][n]$ is set to 1 if the user m has visited the web page n at time stamp i(during time stamp $i-1$ and i), and 0 otherwise. Note that each row of D_i may contains several 1 s since a user may visit more than one web page for a period of time in reality. The server then publishes the statistics of the visits for each page at time stamps i. Here, we define the statistic of the visits for each web page as a query function Q on D_i, $Q(D_i) = X_i = (x_i^1, x_i^2, \cdots, x_i^d)$, where d denotes the total number of pages and x_i^j denotes the number of visits of page j. Since each user can visit several web pages for each time stamp, the sensitivity $\Delta(Q)$ may be large and consequently leads to a huge injected noise.

Instead of directly releasing x_i^j with high privacy leakage, the server publishes a sanitized version of x_i^j, denoted by r_i^j. At time stamp i, based on the statistics X_i, we denote its corresponding sanitized version by $R_i = (r_i^1, r_i^2, \cdots, r_i^d)$. Therefore, the goal of this paper is to design a privacy protection mechanism to generate and publish the sanitized version R_i in real-time and guarantee that the subsequent releases $\mathbf{R} = \{R_1, R_2, ..., R_i, ...\}$ is satisfying w-event privacy.

Here, we briefly explain several important notions to be used throughout the paper.

Utility. The utility of the published data measures how valid the data is used for subsequent analysis or mining tasks. In this paper, we evaluate the utility with the following metrics: Mean Absolute Error (MAE), Mean Relative Error (MRE), and Top-K mining precision.

Sampling Point and Non-Sampling Point. A sampling point is a selected time stamp where the raw statistic is queried and perturbed. The statistic at a non-sampling point will not be queried but instead will be approximated by the perturbed data at last sampling point.

2.2 Related Work

In [4,21], it has been shown that there exist severe privacy risks when users' data is released, and many privacy-assured data publishing schemes have been designed accordingly.

To publish search logs or web browsing data, many schemes were proposed to achieve k-anonymity [2,16]. However, it was shown in [15] that the existing solutions always assume the attackers have no background knowledge, and this is not true in practice. In comparison, the notion of differential privacy proposed in [9] can ensure much stronger privacy guarantee, where a user's privacy can be well protected even if the attackers have obtained the others' information in the database. Following the differential privacy model, [10] proposed the first differentially private scheme called Laplace mechanism. On top of that, many schemes have been presented for achieving differentially private data publishing in the past years.

In [8,20,24], several mechanisms were proposed for the release of statistical data computed based on the static database. Until recently, researchers began to consider releasing time series data. One direction is to study the off-line data release [1,23] while the other direction is to investigate the real-time data publishing [7,11]. The key difference between these two direction is that the solutions for the off-line data release deal with the whole time series data at one time, but the solutions for the real-time data publishing deal with the data streamingly.

In [7,11], the authors proposed differentially private solutions for continual counting queries over time series data, and the techniques can be used for real-time monitoring. Their limitation is that only *event-level* privacy guarantee is provided. That is, only one's presence at a single time stamp is fully protected over the whole data stream.

In [12,13,18], the proposed solutions considered differentially private release of real-time time series. Different from [7,11,13] established a new framework called FAST. FAST consists of sampling and filtering operations with the appealing property of providing *user-level* privacy. That is, the presence of a user over the whole time series is protected. But FAST has the limitation of pre-assigning the maximum times of publications, so it is only suitable for finite-time data publishing. To fill the gap between *event-level* privacy and *user-level* privacy, Kellaris et al. [18] proposed *w-event ϵ-differential privacy*, and it can protect any sequence of events existing in any continuous time stamps of length w over infinite time. Due to its nice property, in this paper we use it to protect users' web browsing traces within any window of w continuous time stamps.

Fan and Xiong [12] took the first step towards sharing web browsing data with differential privacy. They proposed two algorithms based on FAST. The first algorithm slightly changes FAST to the web browsing scenario, which is called univariate Kalman filter (U-KF). The second algorithm, called multivariate Kalman filter (M-KF), establishes a multivariate model and utilizes the Markov property of web browsing behavior. M-KF uses Markov chain to improve accuracy of the prediction step in Kalman filter and have a more accurate result than U-KF. However, the Markov model must be learned by an appropriate training set in advance and the multiple steps of matrix operations in M-KF extremely reduce its efficiency, which is especially vital for a real-time algorithm.

Competitors. We identify three competitors. The first is an application of LPA [10] on w-event privacy, which is also called UNIFORM in [18]. UNIFORM

assigns $\frac{\epsilon}{w}$ to every time stamp, where ϵ is the total budget. And then UNIFORM straightforwardly applies LPA at each time stamp. Obviously, the budget for each time stamp will be very small if w is large, which leads to a very bad utility.

The last two competitors are U-KF and M-KF. Since U-KF and M-KF are not w-event private, we slightly change them to satisfy w-event privacy according to [18] and name the new schemes as U-KF$_w$ and M-KF$_w$. To be precise, we make an instantiation of the two methods which consist of sub mechanisms, each operating on a disjoint w time stamps. In order to guarantee that the total budget allocated in any w successive time stamps is less than ϵ, for each sub mechanism, we allocate budget $\epsilon/2$ to satisfy w-event privacy.

3 SecWeb: Real-Time Web Browsing Data Publishing with Privacy Preservation

In this section, we present our SecWeb design to achieve real-time web browsing data publishing with privacy preservation. In order to realize this purpose, we propose a framework for SecWeb, as shown in Fig. 2, which is mainly composed of five components: *adaptive sampling*, *dynamic grouping*, *adaptive budget allocation*, *grouping based perturbation*, *filtering* and *pre-sampling*.

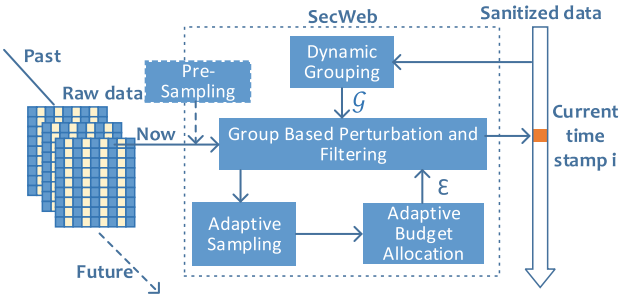


Fig. 2. The framework of SecWeb

Specifically, the *adaptive sampling* component can adjust the sampling rate based on dynamic data, and it enables SecWeb to perturb statistics at selected sampling time stamps while approximating the non-sampled statistics with perturbed statistics at the last sampling time stamp. The *adaptive budget allocation* component can dynamically allocate appropriate budget for each sampling page according to the changing trend of each web page. For the sampling pages at each time stamp, the *dynamic grouping* component can group sampling pages with similar features together, and the *group based perturbation* component can inject Laplace noise to the groups other than individual web pages with the allocated budget to reduce the perturbation error to each web page. Moreover, following FAST, the *filtering* component is used to further enhance the utility

of published sanitized data. Finally, the server publishes the sanitized data after filtering and chooses the new sampling interval for the sampling pages using the *adaptive sampling* component. After presenting SecWeb, we further propose a pre-sampling method to reduce the high query sensitivity, and this method can be integrating with SecWeb seamlessly.

3.1 Adaptive Sampling

Every noisy data release comes at the cost of budget consumption while the entire budget ϵ is a constant. Thus, publishing noisy data at every time stamp will introduce large magnitude of noise when the window size w is large. An efficient way to overcome this problem is to use a sampling mechanism which queries and perturbs statistics at selected time stamps and approximates the non-sampled statistics with perturbed sampled statistics. Consequently, non-sampled statistics can be approximated without any budget allocation, and more budget can be allocated to sampling points within any successive w time stamps given a fixed ϵ.

Figure 3 shows the general idea of the *adaptive sampling* mechanism. The blue line with markers represents the raw data series, the red line denotes the released data and the dashed green lines denote the sampling points. Note that here we inject Laplace noise at each sampling point with value of zero for simplicity. As shown in Fig. 3, the *adaptive sampling* mechanism only samples three points through time stamp 1 to 10, since the raw series change gently and the non-sampled statistics could be roughly approximated. The sampling rate increases when the raw data changes dramatically through time stamp 10 to 20 in order to avoid large error introduced by approximation.

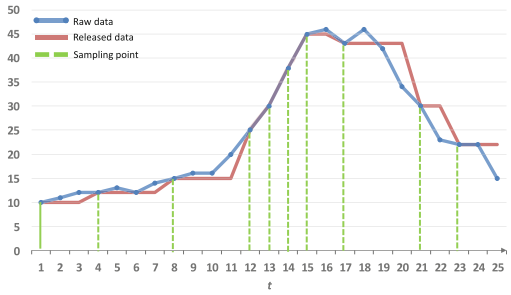


Fig. 3. An illustration of adaptive sampling (Color figure online)

In this paper, we consider both the data dynamics and the remaining budget to design an adaptive sampling mechanism. Specifically, the proportional-integral-derivative (PID) control is utilized to characterize the dynamic data. At the next time stamp, we then choose the next sampling interval for every page with the PID error and the remaining budget. In comparison to FAST [13], here

a new *feedback error* measure is used to compute the PID error. This is because FAST's *feedback error* can be too sensitive to data dynamics, and the adaptive sampling performance would be affected when we have small data values.

Let k_n and k_{n-1} be the current and the last sampling points, respectively. For page j, we have the *feedback error*:

$$E^j_{k_n} = |r^j_{k_n} - r^j_{k_{n-1}}|.$$

It is actually the error of the released data values between the current and the last sampling points. The PID error δ^j for statistics on the jth column of D_{k_n} (i.e., page j) is computed as

$$\delta^j = K_p E^j_{k_n} + K_i \frac{\sum_{o=n-\pi-1}^{n} E^j_{k_o}}{\pi} + K_d \frac{E^j_{k_n}}{k_n - k_{n-1}}, \tag{6}$$

where K_p, K_i, and K_d denote the standard PID scale factors, which respectively represents the proportional gain, the integral gain and the derivative gain. The first term $K_p E^j_{k_n}$ is the proportional error standing for the present error; the second term $K_i \frac{\sum_{o=n-\pi-1}^{n} E^j_{k_o}}{\pi}$ is the integral error standing for the accumulation of past errors, and π is how many recent errors are taken for the integral error; the third term $K_d \frac{E^j_{k_n}}{k_n - k_{n-1}}$ denotes the derivative error used to predict the future error. In our experiments, for the PID controller we choose $\pi = 3$, $K_p = 0.9$, $K_i = 0.1$, and $K_d = 0$.

It may seem that we should choose a small sampling interval if the data rapidly changes. However, this is not always the case. When we have a very small remaining budget, sampling and perturbing statistics at the next time stamp may incur quite high perturbation error. So, a better choice is to adopt a relatively large sampling interval, then the previously-allocated budget can be used again, and it will approximate the statistics at the next time stamp with the previous publication. We have the new sampling interval

$$I = \max\{1, I_l + \theta(1 - (\frac{\delta^j}{\lambda_r})^2)\}, \tag{7}$$

where I and I_l respectively denotes the next and the last sampling intervals of page j. In our settings, $\lambda_r = 1/\epsilon_r$ is chosen to measure the scale of Laplace noise. Here ϵ_r denotes the remaining budget at the next time stamp, and θ denotes a pre-defined scale factor used to adjust the sampling interval. In our experiments, we set $\theta = 10$. In particular, the relative value of PID error δ^j and the scale of Laplace noise λ_r are used to determine the increase or decrease of the sampling interval. In fact, we increase the sampling interval when $\delta^j < \lambda_r$ and decrease it when $\delta^j > \lambda_r$.

3.2 Dynamic Grouping

Intuitively, directly injecting Laplace noise to each statistic is the simple and straightforward way to achieve differential privacy. However, this is not true.

For the web pages with small statistics, their utilities will be severely affected when the privacy level is satisfied, especially when the limited privacy budget should be allocated to multiple time stamps. Even a small noise may cause large relative error when the statistics of sampling pages are small.

Fan et al. [14] proposed a grouping mechanism to solve this kind of problems. The main idea is to aggregate the statistics of similar regions together and inject noise to each aggregated group, and then average the noisy count to each group member. Note that the proposed grouping mechanism in [14] is based on the assumption that the statistics of regions which are close in space have similar changing trend, and the grouping process is performed offline at one time. This assumption however does not hold for real-time web browsing data publishing since the statistics of web pages behave high dynamics and should not be grouped offline at one time.

Inspired by the grouping mechanism in [14], in this paper, we propose a dynamic grouping mechanism that aggregates the web pages with small statistics dynamically based on their real-time changing trend. The main idea is that web pages with small statistics can be grouped together if their statistics are close and the changing trends of statistics are similar.

To realize this objective, we use the released statistics at previous sampling points to predict the statistic at current sampling point as well as characterize the changing trends of statistics. Let $(r^j_{k_{i-\kappa}}, r^j_{k_{i-\kappa+1}}, \cdots, r^j_{k_{i-1}})$ denote the released statistics at previous κ sampling points, and $\bar{x}^j_{k_i}$ denote the predicted statistic at sampling point k_i for page j. We let $\bar{x}^j_{k_i} = \sum_{o=i-\kappa}^{i-1} r^j_{k_o}/\kappa$, and adopt Pearson Correlation Coefficient [22], the most commonly used measure of correlation in statistics, to measure the similarity of changing trend of statistics. Finally, pages with small statistics and high similarity are grouped together.

The pseudocode of the *dynamic grouping* mechanism is formally presented in Algorithm 1. Note that at each time stamp, dynamic grouping only considers the set of pages that need to be sampled, denoted by Ψ. Let $\mathcal{G}_{k_i}$ denote the group strategy at time stamp k_i. First, the mechanism predicts the statistic at k_i for each sampling page in Ψ. Let τ_1 denotes the noise resistance threshold that reflects whether the statistics of pages have sufficient capacity to resist noise. If $\bar{x}^j_{k_i} \geq \tau_1$, the page itself can be a group; otherwise, the page is encouraged to be grouped with other pages. Thus, in lines 2–7, the mechanism filters out the pages that can resist noise individually which do not need to be grouped with other pages together. These found pages are put into the group strategy $\mathcal{G}_{k_i}$ where each of them is an individual group.

Lines 8–20 describes how to group web pages with small statistics together. Generally speaking, two pages i and j can be grouped together if they have small error between $\bar{x}^i_{k_i}$ and $\bar{x}^j_{k_i}$, and also they have sufficient similarity of the changing trend. The similarity of two pages i and j at time stamp k_i can be calculated by the Pearson Correlation Coefficient of R^i_k and R^j_k. Let τ_2 denote the similarity threshold that decides whether two pages have similar changing trends of statistics. Thus, when the similarity of two pages is no less than τ_2, the two pages have sufficient similarity. Let τ_3 denote the error threshold that

Algorithm 1. Dynamic Grouping

Input: Ψ: the collection of sampling pages;
$R_k^j = (r_{k_{i-\kappa}}^j, r_{k_{i-\kappa+1}}^j, \cdots, r_{k_{i-1}}^j)$: the released statistics at previous κ sampling points for a sampling page j.
Output: Group strategy $\mathcal{G}_{k_i}$.
1: Calculate $\bar{x}_{k_i}^j = \sum_{o=i-\kappa}^{i-1} r_{k_o}^j / \kappa$ for each page j in Ψ
2: **for** each page in Ψ, say j **do**
3: **if** $\bar{x}_{k_i}^j > \tau_1$ **then**
4: Let the page j itself as a group and add it to $\mathcal{G}_{k_i}$
5: Remove page j from Ψ
6: **end if**
7: **end for**
8: Sort Ψ in increasing order according to $\bar{x}_{k_i}^j$
9: **while** $\Psi \neq \varnothing$ **do**
10: Initialize a empty group g with the first page in Ψ
11: Let $o = 2$
12: **while** $o < \Psi.length$, $\bar{x}_{k_i}^o - \bar{x}_{k_i}^1 < \tau_2$ and the sum of $\bar{x}_{k_i}^j$ in $g < \tau_1$ **do**
13: $pc \leftarrow$ calculate Pearson Correlation Coefficient between page o and page 1
14: **if** $pc > \tau_3$ **then**
15: Add page o to g
16: **end if**
17: $o = o + 1$
18: **end while**
19: Remove the pages in g from Ψ and add g to $\mathcal{G}_{k_i}$
20: **end while**
21: Return grouping strategy $\mathcal{G}_{k_i}$

decides whether two pages are close or not in terms of predicted statistics. Thus, when the error is less than τ_3 and the similarity is larger than τ_2, two pages are encouraged to be grouped together.

In line 8, the dynamic grouping mechanism first sorts the remaining web pages in Ψ in increasing order according to $\bar{x}_{k_i}^j$. In lines 9–20, the mechanism repeatedly forms groups by putting the pages in Ψ with small error and high similarity to the first page in Ψ, and puts the formed groups into the group strategy $\mathcal{G}_{k_i}$. The process terminates until there is no page in Ψ. Note that when forming a group, the grouping process checks the remaining pages one by one in Ψ and put qualified pages into a group. However, if the sum of predicted statistics of all pages put in the group is larger than τ_1, which means the group has sufficient capacity to resist noise, no more page need to be added to this group and a new grouping process can start.

3.3 Adaptive Budget Allocation

To achieve w-event differential privacy, we should make sure that the budget sum of any successive w time stamps is at most ϵ. Here, we propose an adaptive budget allocation mechanism based on the trend of data change to adaptively

Algorithm 2. Adaptive Budget Allocation

Input: Privacy budget ϵ, new sampling interval I, allocated budget for each time stamp $(\epsilon_1, ..., \epsilon_{i-1})$, and the maximum allocated budget at each sampling point ϵ_{max}. Note that $\epsilon_k = 0$ if time stamp k is not a sampling point.

Output: Budget allocation ϵ_i for the sampling time stamp i

1: Compute the remaining budget $\epsilon_r = \epsilon - \sum_{k=i-w+1}^{i-1} \epsilon_k$
2: Compute the portion $p = \min(\phi \cdot \ln(I+1), p_{max})$
3: Compute the allocated budget $\epsilon_i = \min(p \cdot \epsilon_r, \epsilon_{max})$

allocate appropriate budget at *each sampling point*. In our design, based on the data change trend, we adjust the length of the sampling interval. In fact, when data changes rapidly (slowly) the new sampling interval is small (large). Thus, for the small sampling interval, we could infer that the data is changing rapidly, so we have more sampling points within a time window of length w. Then, we determine to put a small portion of the remaining budget to the next sampling point. In this case, more available budget can be given to the (potential) successive sampling points. When we have a large the sampling interval, we could infer that the data is changing slowly, so we only have fewer sampling points within the time window of length w. So, we determine to put a large portion of the remaining budget to the next sampling point.

To achieve our goal, we propose to use the natural logarithm to link p (the portion of the remaining budget) and I. So, we define $p = \phi \cdot \ln(I+1)$, where the scale factor ϕ ranges in (0,1]. Because I has the minimum value 1, to avoid the case that $p = 0$ we use $\ln(I+1)$ other than $\ln I$.

Algorithm 2 formally presents the adaptive budget allocation mechanism. First, we compute the remaining budget ϵ_r in $[i-w+1, i]$. Here ϵ_r equals to ϵ minus the budget sum allocated in $[i-w+1, i-1]$. Then we compute the portion p to determine how much budget will be used for the current sampling point i. It is worth noting that $p \leq p_{max}$ is set to avoid the case that we leave to the next sampling point too few budget. Finally, we compute the budget allocated to the current time stamp as $\epsilon_i = \min(p \cdot \epsilon_r, \epsilon_{max})$, where ϵ_{max} is the upperbound for the budget allocated at each sampling point. The introduction of ϵ_{max} is due to the fact that the utility enhancement is small when the allocated budget is larger than ϵ_{max}, say $\epsilon_{max} = 0.2$ when $\epsilon = 1$.

3.4 Group-Based Perturbation and Filtering

At each time stamp, we apply Laplace mechanism to inject Laplace noise to statistics at sampling pages to provide differential privacy guarantee. For each non-sampling page, the publication is approximated by its last release. Here, Laplace mechanism is applied to every group other than every page. Then we compute the average of the perturbed statistic to each page. To guarantee that the total budget assigned to every page at any successive w time stamps is less than ϵ, the budget assigned to a group is the smallest budget assigned to pages in the group.

Assume we have a group g of φ pages. Thus, g contains φ columns of D_i and $g \subseteq D_i$. We use $f(g)$ to denote the statistic function that accumulates all 1's in g. We use $\lambda(g)$ to denote the scale of Laplace noise injected to $f(g)$. The Laplace mechanism is applied to group g, and we have

$$\begin{aligned}\mathcal{M}(g) &= f(g) + Lap(\lambda(g)) \\ &= \sum_{j=1}^{\varphi} \sum g[j] + Lap(\Delta(f)/\min(\epsilon_{g[j]})),\end{aligned} \tag{8}$$

where $g[j]$ is the jth column of g and $\Delta(f)$ is decided by the database.

Then the perturbed statistic for each column/page at group g is calculated as the average of $\mathcal{M}(g)$. That is,

$$\mathcal{M}(g[j]) = \mathcal{M}(g)/\varphi, \quad \forall j = 1, \cdots, \varphi. \tag{9}$$

However, we would not release $\mathcal{M}(g[j])$ directly and further apply the Kalman filtering mechanism of FAST algorithm [13] to improve the utility of released statistics. The detail of the mechanism can be found in [13].

3.5 Pre-sampling to Reduce Sensitivity

We use dynamic grouping to diminish the perturbation error on small statistics, which greatly improves the data utility of pages with small counts. However, the high query sensitivity will still bring a huge injected noise, which may also have a bad influence on the utility of released data.

In [17], the authors proposed a sampling method to generate a small portion of the original database, which is used to calculate the grouping strategy. Inspired by their work, we consider whether we can further reduce the injected noise by cutting down the sensitivity $\Delta(Q)$ through a pre-sampling method.

Here, we propose a concise and effective pre-sampling method to generate the representative database D'_i at each sampling point i. Specifically, at each sampling point i, our method gets a new database D'_i by randomly sampling m 1s in each row in D_i(if there are only $n < m$ 1s in a row, preserve them all), and setting the remaining to 0. Consequently, there are at most m 1s in each row after pre-sampling, i.e., each user can visits at most m web pages per time stamp and the sensitivity $\Delta(Q(D'_i)) = m$. We then use D'_i to replace the original database D_i, and the remaining procedures are just the same as the original SecWeb. The only difference is, for each group g, only $Lap(m/\min(\epsilon_g))$ of noise is needed to be injected, where m can be a user-defined parameter. Note that the pre-sampling method also cause a biased estimate error since D'_i is a selected portion of D_i. The error is heavily data-dependent which can not be rigorously analyzed, and consequently we cannot give a certain value of m to get the optimal performance without knowing the data distribution. Intuitively, the value of m should be close to the average counts of each page in the database, we will test the effectiveness of pre-sampling over different values of m in our experiments.

3.6 Privacy Analysis

Theorem 3. *SecWeb satisfies w-event ϵ-differential privacy.*

Proof. According to Axiom 2.1.1 in [19], post-processing the sanitized data maintains privacy as long as the post-processing algorithm does not use the sensitive information directly. In SecWeb, group-based perturbation is the only component processing the raw data directly, while other components process the sanitized data. Thus, we will first show that the group-based perturbation component achieves w-event ϵ-differential privacy, then it is easy to prove that SecWeb can achieve the same privacy guarantee.

Based on $\mathcal{G}$, D_i is separated to n disjoint groups $\{g_1, g_2, \ldots, g_n\}$, and every group contains some columns of D_i. Without any loss of generality, we consider g_1, and suppose g_1 consists of φ_1 columns. Based on Eq. 8, we have

$$\begin{aligned}\mathcal{M}(g_1) &= f(g_1) + Lap(\lambda(g_1)) \\ &= \sum_{j=1}^{\varphi_1} \sum g_1[j] + Lap(\Delta(f)/\min(\epsilon_{g_1})).\end{aligned}$$

Here, $g_1[j]$ denotes g_1's j-th column.

Based on Theorem 1, $\mathcal{M}(g_1)$ achieves $\min(\epsilon_{g_1})$-differential privacy. Based on Axiom 2.1.1 in [19], $\mathcal{M}(g_1[j])$ $(\forall j = 1, \cdots, \varphi)$ also achieves $\min(\epsilon_{g_1})$-differential privacy. Analogously, every group runs Laplace mechanism independently on a column/page in group g_k satisfying $\min(\epsilon_{g_k})$-differential privacy. We use $\hat{\epsilon}_k$ and ϵ_k to respectively denote the budget used for perturbation and the allocated budget (generated by the adaptive budget allocation component) for a page at time stamp k, then we have $\hat{\epsilon}_k \leq \epsilon_k$.

Based on Theorem 2, to show that the perturbation component for a page achieves w-event ϵ-differential privacy, we should show that for each t and $i \in [t]$, $\sum_{k=i-w+1}^{i} \hat{\epsilon}_k \leq \epsilon$ will hold. Because our adaptive budget allocation component guarantees that $\sum_{k=i-w+1}^{i} \epsilon_k \leq \epsilon$ for any w successive time stamps, and $\hat{\epsilon}_k \leq \epsilon_k$, then we have $\sum_{k=i-w+1}^{i} \hat{\epsilon}_k \leq \epsilon$. Hence, the perturbation component on each group achieves w-event ϵ-differential privacy. Hence, SecWeb can also achieve the same privacy guarantee.

Theorem 4. *SecWeb with pre-sampling(SecWeb-S for short) satisfies w-event ϵ-differential privacy.*

Proof. The only differences between SecWeb-S and SecWeb are the procedure of pre-sampling and the noise injected in group-based perturbation. Consider a possible D_i' derived from pre-sampling D_i, each row in D_i' contains at most m 1s. Therefore, the sensitivity of the query f on D_i' in proof Sect. 3.6 is $\Delta(f(D_i')) = m$. Recall that, for each group g we inject noise $Lap(m/\min(\epsilon_g))$, where $\min(\epsilon_g)$ is the minimum budget in g, thus we can derive that the pre-sampling and group-based perturbation in SecWeb-S satisfies $\min(\epsilon_g)$-differential privacy. Similar to proof Sect. 3.6, we can then conclude that SecWeb-S satisfies w-event ϵ-differential privacy.

4 Performance Evaluation

In this section, we used a real-world web dataset, WorldCup [3], as a source of the input stream to evaluate the performance of SecWeb. The entire dataset contains 1,352,804,107 web server logs collected by the FIFA 1998 World Cup Web site between April 30, 1998 and July 26, 1998. These logs are the requests made to 89,997 different URLs and each log consists of a client ID, a requested URL, a time stamp, etc. We randomly choose 1,500 URLs as the test set, create a stream from the set and publish the data per hour, which has a total of 1000 time stamps. The query sensitivity defined in Sect. 2.1 is 30, and the average count of each page per time stamp is 17.9.

We compare our schemes SecWeb and SecWeb-S with three competitors as introduced in Sect. 2.2, UNIFORM, U-KF$_w$ and M-KF$_w$, where the latter are the first two schemes proposed for web browsing monitoring with differential privacy [12]. All the mechanisms are fine-tuned and implemented in Python. We conduct all the experiments on a machine with Intel Core i5 CPU 2.9 GHz and 12 GB RAM, running Windows 10. We set $\phi = 0.2$ for the adaptive budget allocation, and let $\tau_1 = 50$, $\tau_2 = 0.5$ and $\tau_3 = 25$ for dynamic grouping.

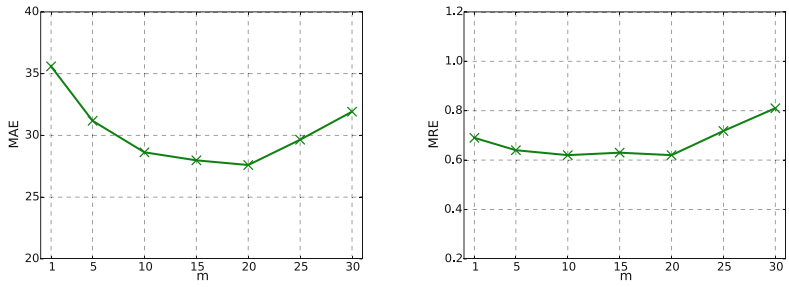
We use Mean Absolute Error (MAE) and Mean Relative Error (MRE) as the utility metrics to evaluate the performance of the five mechanisms.

For any web page, let $\mathbf{x} = \{x_1, ..., x_n\}$ denote the raw time series and $\mathbf{r} = \{r_1, ..., r_n\}$ denote the sanitized time series. The MAE and MRE for this page are

$$\mathbf{MAE}(\mathbf{x}, \mathbf{r}) = \frac{1}{n} \sum_{i=1}^{n} |r_i - x_i| \tag{10}$$

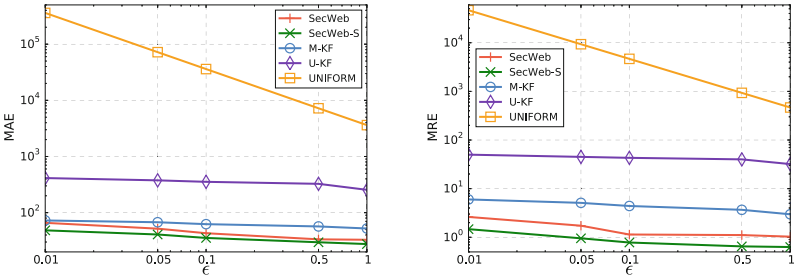
$$\mathbf{MRE}(\mathbf{x}, \mathbf{r}) = \frac{1}{n} \sum_{i=1}^{n} \frac{|r_i - x_i|}{\max(\gamma, x_i)} \tag{11}$$

For the bound γ, we set its value to 0.1% of $\sum_{i=1}^{n} x_i$ to mitigate the effect of excessively small results. In experiments, we first calculate the MAE and MRE for each page and then figure out the average of all pages as the final results.



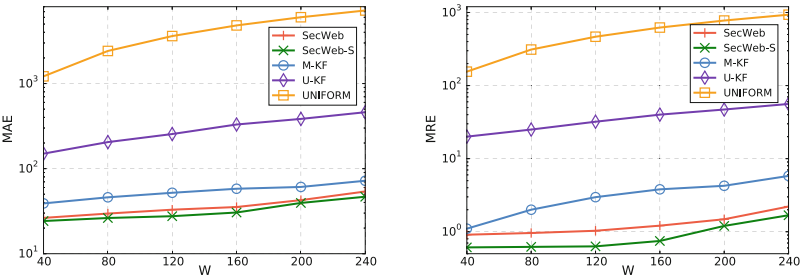
(a) MAE (b) MRE

Fig. 4. Utility comparison when m changes ($w = 120, \epsilon = 1$)



(a) MAE (b) MRE

Fig. 5. Utility comparison when ϵ changes ($w = 120$)



(a) MAE (b) MRE

Fig. 6. Utility comparison when w changes ($\epsilon = 1$)

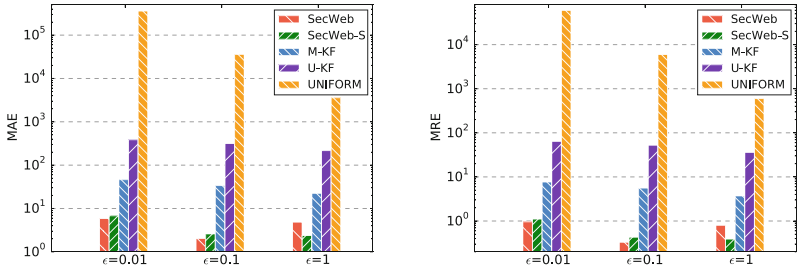
Varying parameter m. Figure 4 illustrates the different performances of SecWeb-S when changing the value of m. As we can see, the pre-sampling method can improve the data utility by reducing the query sensitivity, and SecWeb-S achieves the best performance in both MAE and MRE when $m = 20$. This results also verify our intuition that the value of m should be close to the average count of each page per time stamp(the average count is 17.9 in our dataset). We set $m = 20$ for SecWeb-S in the rest of our experiments.

Utility vs. Privacy. Figure 5 shows the relationship between data utility and privacy budget ϵ. As we can see, the MAE and MRE of all five mechanisms decrease when ϵ increases. This is because that lager ϵ requires smaller noise to preserve privacy, which results in a better utility. UNIFORM has the worst performance since it uniformly allocates the budget and simply adopts LPA at each time stamp. U-KF$_w$ performs much better compared to UNIFORM, since the posteriori estimate on each web page produced by Kalman Filter extremely improves the utility. While the improved method M-KF$_w$ performs better than U-KF$_w$ due to its adoption of fisrt-order Markov chain utilizing user pattern to improve utility. SecWeb and SecWeb-S have the best performance compared to other algorithms, and SecWeb-S performs a little better. The reason is that the

well designed dynamic grouping strategy significantly improves the capacity of resisting Laplace noise for pages with small statistics by grouping them together, and the adaptive sampling mechanism also helps avoiding unnecessary noise. The pre-sampling method in SecWeb-S also helps reducing the injected noise.

Utility vs. w. Figure 6 shows the comparison of different utility metrics between the five schemes when window size w varies from 40 to 240. We can also observe that SecWeb and SecWeb-S outperforms other algorithms when w changes. The MAE and MRE of M-KF$_w$ and U-KF$_w$ increase when w becomes large. The reason is that the budget allocated to each time stamp becomes less when w increases since both of them allocate budget uniformly, which results in larger error. The MAE and MRE of our two schemes are much smaller than that of M-KF$_w$ and U-KF$_w$ and are robust to w changes, which is because that SecWeb takes the remaining budget into consideration to adaptively allocate budget on sampling points to reduce the error.

Effects of Dynamic Grouping. We evaluate the performance of our grouping method. Specifically, we calculate the average statistics of each page and pick out the half part of pages with smallest statistics being the test set to see the performance of dynamic grouping. Figure 7 shows the comparison of MAE and MRE between the five schemes on the pages with small statistics. We can observe that SecWeb achieves a much better utility on both MAE and MRE. The reason is that the grouping mechanism in SecWeb groups these pages together dynamically, injects noise to the whole group and averages the counts, which can extremely reduce the perturbation error compared to the schemes that inject noise to each page individually. Note that SecWeb-S also achieves a much better utility, but not always as good as SecWeb. That is because the pages that we select have small counts, where the selected portion of the original database produced by pre-sampling mechanism cannot represent them well since these pages have a less times to be selected than the pages with larger statistics.



(a) MAE (b) MRE

Fig. 7. Utility comparison on pages with small statistics($w = 120, \epsilon = 1$)

Running Time. Table 2 shows the comparison of time complexity of the five mechanisms. We can see that U-KF$_w$ and UNIFORM are the fastest mechanisms

Table 2. Comparison of running time

	UNIFORM	U-KF$_w$	M-KF$_w$	SecWeb	SecWeb-S
Time complexity	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(d^3)$	$\mathcal{O}(d^2)$	$\mathcal{O}(d^2)$
Running time (d = 1500)	0.4×10^{-5} s	0.4×10^{-5} s	1.2 s	0.4×10^{-2} s	0.45×10^{-2} s

with time complexity $\mathcal{O}(d)$, while M-KF$_w$ is the slowest mechanism with time complexity $\mathcal{O}(d^3)$, where d is the number of pages. SecWeb and SecWeb-S with the time complexity of $\mathcal{O}(d^2)$ are slower than U-KF$_w$ but much faster than M-KF$_w$. Note that although U-KF$_w$ and UNIFORM are the most fast schemes, they have the worst utility as seen from Figs. 5 and 6. Although the MAE and MRE of M-KF$_w$ are close to that of SecWeb, SecWeb is much faster than M-KF$_w$. Note that SecWeb-S is a little bit slower than SecWeb since it has a pre-sampling procedure. Overall, SecWeb and SecWeb-S achieve a well tradeoff between time efficiency and utility.

5 Conclusions

In this paper, we proposed SecWeb to enable continually publishing aggregated web browsing data for real-time monitoring purposes with w-event privacy guarantee. SecWeb is designed with five integrated components: *adaptive sampling*, *adaptive budget allocation*, *dynamic grouping*, *group-based perturbation* and *filtering*. We proved that SecWeb satisfies w-event ϵ-differential privacy. We further proposed a pre-sampling method to reduce the high query sensitivity and integrated it with SecWeb seamlessly (SecWeb-S). Extensive experiments on real-world dataset showed that SecWeb-S outperforms the existing methods and improves the utility of the released data with strong privacy guarantee.

Acknowledgment. Qian and Zhibo's research is supported in part by National Natural Science Foundation of China (Grant No. 61373167, 61502352), National Basic Research Program of China (Grant No. 2014CB340600), Wuhan Science and Technology Bureau (Grant No. 2015010101010020), and Natural Science Foundation of Hubei Province and Jiangsu Province (Grant No. 2015CFB203, BK20150383). Kui's research is supported in part by US National Science Foundation under grant CNS-1262277. Qian Wang is the corresponding author.

References

1. Acs, G., Castelluccia, C.: A case study: privacy preserving release of spatio-temporal density in Paris. In: Proceedings of ACM SIGKDD, pp. 1679–1688 (2014)
2. Adar, E.: User 4xxxxx9: Anonymizing query logs. In: Proceedings of Query Log Analysis Workshop, International Conference on World Wide Web (2007)
3. Arlitt, M., Jin, T.: Workload characterization of the 1998 world cup web site. Technical report, HPL-1999-35R1, HP (1999)

4. Barbaro, M., Zeller, T.: A face is exposed for AOL searcher no. 4417749. The New York Times (2006)
5. Canali, D., Balzarotti, D.: Behind the scenes of online attacks: an analysis of exploitation behaviors on the web. In: 20th Annual Network & Distributed System Security Symposium (NDSS 2013) (2013)
6. Chan, T.H.H., Li, M., Shi, E., Xu, W.: Differentially private continual monitoring of heavy hitters from distributed streams. In: Proceedings of Privacy Enhancing Technologies, pp. 140–159 (2012)
7. Chan, T.H.H., Shi, E., Song, D.: Private and continual release of statistics. ACM Trans. Inf. Syst. Secur. **14**(3), 26 (2011)
8. Cormode, G., Procopiuc, M., Srivastava, D., Tran, T.T.: Differentially private publication of sparse data. arXiv preprint arXiv:1103.0825 (2011)
9. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006). doi:10.1007/11787006_1
10. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Proceedings of Theory of Cryptography, pp. 265–284 (2006)
11. Dwork, C., Naor, M., Pitassi, T., Rothblum, G.N.: Differential privacy under continual observation. In: Proceedings of ACM STOC, pp. 715–724 (2010)
12. Fan, L., Bonomi, L., Xiong, L., Sunderam, V.: Monitoring web browsing behavior with differential privacy. In: Proceedings of ACM WWW, pp. 177–188 (2014)
13. Fan, L., Xiong, L.: An adaptive approach to real-time aggregate monitoring with differential privacy. IEEE Trans. Knowl. Data Eng. **26**(9), 2094–2106 (2014)
14. Fan, L., Xiong, L., Sunderam, V.: Differentially private multi-dimensional time series release for traffic monitoring. In: Proceedings of Data and Applications Security and Privacy, pp. 33–48 (2013)
15. Götz, M., Machanavajjhala, A., Wang, G., Xiao, X., Gehrke, J.: Publishing search logsa comparative study of privacy guarantees. IEEE Trans. Knowl. Data Eng. **24**(3), 520–532 (2012)
16. Hong, Y., He, X., Vaidya, J., Adam, N., Atluri, V.: Effective anonymization of query logs. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 1465–1468 (2009)
17. Kellaris, G., Papadopoulos, S.: Practical differential privacy via grouping and smoothing. In: Proceedings of the VLDB Endowment, vol. 6, pp. 301–312. VLDB Endowment (2013)
18. Kellaris, G., Papadopoulos, S., Xiao, X., Papadias, D.: Differentially private event sequences over infinite streams. Proc. VLDB Endowment **7**(12), 1155–1166 (2014)
19. Kifer, D., Lin, B.R.: Towards an axiomatization of statistical privacy and utility. In: Proceedings of ACM PODS, pp. 147–158 (2010)
20. Korolova, A., Kenthapadi, K., Mishra, N., Ntoulas, A.: Releasing search queries and clicks privately. In: Proceedings of the 18th International Conference on World Wide Web, pp. 171–180 (2009)
21. Olejnik, L., Castelluccia, C., Janc, A.: Why Johnny can't browse in peace: on the uniqueness of web browsing history patterns. In: 5th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2012) (2012)
22. Pearson, K.: Note on regression and inheritance in the case of two parents. Proc. R. Soc. Lond. **58**, 240–242 (1895)
23. Rastogi, V., Nath, S.: Differentially private aggregation of distributed time-series with transformation and encryption. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, pp. 735–746 (2010)

24. Xu, J., Zhang, Z., Xiao, X., Yang, Y., Yu, G., Winslett, M.: Differentially private histogram publication. VLDB J. Int. J. Very Large Databases **22**(6), 797–822 (2013)
25. Zang, H., Bolot, J.: Anonymization of location data does not work: a large-scale measurement study. In: Proceedings of ACM MobiCom, pp. 145–156 (2011)