Swellfish Privacy: Publishing Continuous Private Sum Aggregates with High Utility

Christine Tex
Karlsruhe Institute of Technology
christine.tex@kit.edu

Martin Schäler
Karlsruhe Institute of Technology
martin.schaeler@kit.edu

Klemens Böhm
Karlsruhe Institute of Technology
klemens.boehm@kit.edu

***Abstract*—Continuous monitoring of personal data like power consumption is common. To mitigate individual bias, only aggregated statistics like the `Sum` of the power consumption values of several households tend to be monitored. As aggregates nevertheless contain private information, one sanitizes continuous aggregates with the w-event differential privacy framework. However, this framework does not allow for privacy parameters that vary over time, and highly-sensitive aggregate queries like `Sum`. Both issues lead to poor data utility when publishing `Sum` aggregates continuously. We address both issues by proposing Swellfish Privacy. It allows for temporally varying privacy parameters. This also helps to control the high sensitivity of the `Sum` query. In experiments with streams of real-world power consumption data and state-of-the-art competitors, including approximate variants, we show that Swellfish Privacy improves error rates compared to the competitors by an order of magnitude.**

I. Introduction

The digitization of every-day live goes along with the availability of many streams of personal data. Examples are locations continuously recorded by smart phones, or power consumptions recorded with smart meters. Monitoring these streams continuously is expected to lead to new insights on how humans behave. This facilitates numerous new applications [1]. To reduce data volume and to compensate individual bias, one often monitors *aggregated statistics* of the streams, such as `Counts` [2], [3], `Histograms` [4], [5] and `Sums` [6], instead of individual data. However, the disclosure of aggregated statistics may still compromise the privacy of the individuals due to differencing attacks [7]. To avoid this, one sanitizes aggregated streams. Therefore, one typically adds a well-defined amount of noise to the aggregate ensuring a specific privacy level. By concept, sanitizing aggregates is a balancing act between adding enough noise to ensure privacy, and adding as little noise as possible to keep high data utility.

For databases of any type (e.g., static, time series, or streaming data), differential privacy [8] and its extensions [9]–[11] is the current gold standard for aggregate sanitization. The idea is to give a provable statistical indistinguishability guarantee w.r.t. an aggregate query (e.g., `Sum`) for two *neighboring* databases, differing in the data, e.g., locations, of one individual only. That is, with a certain residual risk, it is not possible to decide whether the data of an individual is part of the aggregate.

Scope and Difficulties

In this paper, we focus on differentially private continuous publishing of `Sum` aggregates. Our running example, which is publishing of aggregated power consumption data, motivates this. Here, the aggregates reflect the power consumption of a neighborhood or an entire city. One might use them for instance to predict future power consumption [12]. Offering good data utility for continuous differentially private `Sum` aggregates is notoriously difficult for two reasons we explain shortly. The first difficulty is directly related to the streaming scenario, while the second refers to the applied aggregate query `Sum`. Each difficulty in isolation leads to high noise; in combination their effect is multiplied. Our following explanation of the two difficulties not only illustrates them, but also is the motivation to unveil a potential for noise reduction. To our knowledge, this potential is untapped so far. It goes beyond incremental improvements of existing privacy mechanisms and also is a better fit to real human behavior.

Streaming Scenario: Differential privacy originally ensures *user-level* privacy, as neighboring databases differ in the whole data of one user. As user-level privacy is not implementable for streams due to their infinite length, w-event differential privacy has been proposed [3]. The rationale behind w-event differential privacy is not to protect the whole stream, but any window of at most w consecutive time stamps. So any event of length w or less, like the usage of an appliance, is protected during the whole time while one continuously publishes the aggregates. Therefore, one selects w such that *the longest event* that might ever occur is protected. Naturally, the higher w is, the higher is the noise added to the aggregate. Thus, small values of w are preferable.

`Sum` Query: Most mechanism for w-event differential privacy use the Laplace mechanism, or other sensitivity-based mechanisms, as a building block [2], [3], [5]. This mechanism adds noise to the aggregate which is proportional to its *sensitivity*. The sensitivity is the largest possible difference between the aggregates computed on two neighboring databases. While the sensitivity of, say, the `Count` aggregate query is one, the sensitivity of the `Sum` aggregate query generally is infinite. To limit it, one usually relies on an upper (observed) boundary [6], [13]. For instance, for power consumption data, to determine this bound, one usually relies on the highest power consumption an event can have.

Contributions

As we show in this paper, even when using a state-of-the art w-event differential privacy mechanism with plausible assumptions towards w and the sensitivity, the noise for `Sum` aggregates over power consumption streams remains high. However, our investigations on the problem reveal an so far unconsidered potential for improving data utility. We argue that selecting the window length and sensitivity as described above leads to an unnecessarily high amount of noise in case long and high-power events are sparse during the lifetime of the stream. Further, from the perspective of the individuals, parameter selection is a one-time decision, and the parameters cannot be changed. We observe that this is not in line with common human behavior and leads to a needless conservative parameter selection. So we target at designing a privacy framework, dubbed Swellfish Privacy, and a respective mechanism, which (1) generalizes w-event differential privacy, and (2) allows for temporally varying secret specifications to target the above two difficulties regarding the noise scale. In a nutshell, we make the following contributions:

- We propose Swellfish Privacy, that is a generalization of w-event differential privacy. In contrast to w-event differential privacy, it allows for the flexible specification of secrets that change over time. We give a mechanism that is based on Laplace perturbation satisfying the definition.
- We prove that any w-event differential privacy mechanism gives Swellfish Privacy if the parameters are set accordingly. However, in two experiments using real-world power consumption streams, and state-of-the-art w-event mechanism like RescueDP [5] as competitors, we show that the data utility of these mechanism is an order of magnitude lower than the one of our mechanism. This is because they do not target at the mentioned difficulties.
- As approximate differential privacy is known to improve data utility for high values of w, i.e., long events, in particular [14], we further propose an approximation variant of Swellfish Privacy, together with a mechanism. In the experiments just mentioned, we also compare the approximate variants of the mechanisms with the ”pure“ ones. The primary results are that (1) our ”pure“ mechanism outperforms all ”pure“ and approximate competitors in all experiments, and (2) our approximate mechanism leads to better data utility in case long events must be hidden.

II. Preliminaries

In this section, we introduce preliminaries for this paper. To this end, we first introduce some notation on infinite power consumption streams and the concept of stream prefixes. Then, we recall the ϵ-differential privacy framework for static databases, as foundation of our work. Last, we introduce w-event differential privacy, the current state-of-the-art privacy framework for infinite data streams.

A. Notation on Power Consumption Streams

Let $I = \{1, .., N\}$ be a set of individuals. Every individual is associated to a house or a flat in a city equipped with a smart meter. It measures the power consumption of the individual in kilowatts (kW) in equidistant time intervals of length τ. Unless stated otherwise, we assume $\tau = 15$ minutes, which is the time interval utilized in Germany. A time stamp t is a natural number reflecting the measured power consumption $t{\cdot}\tau$ minutes after the start of the measurements, i.e., installation of the smart meter. For readability, we assume that the smart meters of all individuals have been installed at the same time. We denote the measured power consumption of individual $i \in I$ at time stamp t by D_t^i.

TABLE I
Stream prefix S_3 and Sum aggregation.

Individual	$t = 1$ (1 p.m.)	$t = 2$ (2 p.m.)	$t = 3$ (3 p.m.)
$i = 1$	0.2	0.3	0.2
$i = 2$	0.4	0.4	0.3
...	...	...	
$i = N$	0.0	0.2	0.3
`Sum`	126,259	123,313	119,192

We differentiate between *static* power consumption databases and *continuous streams*. For an arbitrary, but fixed, time stamp t, the vector $D_t = (D_t^1, .., D_t^{|I|})$ that summarizes the consumption of all individuals is a static database, as its values do not change once collected. If we talk about a static database for any arbitrary, but fixed t, we simply write D. In contrast, a continuous stream $S = <D_1, D_2, ..>$ is a sequence of continuous collected static databases with infinite length. Due to their infinite length, we need a notation for static stream snapshots we can work with in this paper. Therefore, we use the notation of *stream prefixes* already introduced in [3]. The prefix $S_{\mathcal{T}} :=< D_1, .., D_{\mathcal{T}} >$ of a stream S consists of all static databases D_t that came up in the stream S until time stamp $\mathcal{T} \geq t$. For an illustration, see Table I. Further, if $\mathcal{J} = [t_B, t_E]$ is an interval, $S_{\mathcal{J}}$ is given by $S_{s.\mathcal{J}} = <D_{t_B}, .., D_{t_E}>$. We assume that missing values in a stream prefix, occurring, e.g., in case of a broken smart meter, are marked by a blank value $\perp$. A summation aggregation $\texttt{Sum}(S_{\mathcal{T}})$ over a stream prefix is

$$\texttt{Sum}(S_{\mathcal{T}}) = <\sum_{i \in I} D_1^i, .., \sum_{i \in I} D_{\mathcal{T}}^i> .$$

In this paper, we aim at publishing $\texttt{Sum}(S_{\mathcal{T}})$ continuously in a differentially private manner.

B. ϵ-Differential Privacy

Differential privacy is the current gold standard of privacy frameworks for a static database D. In this section, we first recall its definition. Then, we recall the Laplace mechanism that is frequently used to sanitize static databases according to the differential privacy framework. Last, we state properties of differential privacy, like the fulfillment of well-established design guidelines [9] for privacy frameworks, we also have to prove for Swellfish Privacy.

1) Definition: Definition 1 contains the definition of differential privacy. It states that if mechanism $\mathcal{M}$, i.e., a sanitized aggregate query, outputs an aggregate R, it is hard to distinguish whether database D or D' was the input for the mechanism. As perfect indistinguishability cannot be achieved [8], one specifies a *privacy level*, also called *privacy budget*, $\epsilon > 0$, that predefines how hard it is to distinguish between the aggregates, The smaller ϵ is, the higher is the level of indistinguishability, i.e., the provided privacy. Frequently, we choose ϵ from the range $(0, 1.0]$.

Definition 1 (Differential Privacy [8])**.** *A randomized mechanism $\mathcal{M}$ gives ϵ-differential privacy if for all neighboring databases D and D', and all $R \in Range(\mathcal{M})$,*

$$Pr[\mathcal{M}(D) = R] \leq e^{\epsilon} \cdot Pr[\mathcal{M}(D') = R].$$

The above definition is based on the notation of neighboring databases. Two static databases are neighbors, if we obtain one from the other by adding, removing or exchanging an individual (*user-level privacy*). In other words, if the Hamming distance d_H between both databases is one, cf. Definition 2.

Definition 2 (Neighboring Databases [8])**.** *Two databases D, D' are neighboring according to the d_H metric, short $(D, D') \in \mathcal{N}_H(1)$, iff $d_H(D, D') = 1$.*

Hence, when considering numeric databases like power consumption databases, literature frequently switches to an L_1-metric-based definition of neighboring databases, cf. Definition 3. There, two databases are neighboring, if their values differ by α. This definition is useful, if we aim at protecting events that have a maximum power consumption of α.

Definition 3 (α-L_1-Neighboring Databases [15])**.** *Two databases D, D' are α-neighboring according to the L_1 metric, short $(D, D') \in \mathcal{N}_1(\alpha)$, iff $d_{L_1}(D, D') = \alpha$.*

2) Laplace Mechanism: Having defined differential privacy, we aim at a mechanism $\mathcal{M}$ that provides it. For numeric databases, the most popular one is the Laplace mechanism. It allows to transform a given aggregate query $\mathtt{Q}$, e.g., $\mathtt{Q} = \mathtt{Sum}$, into a randomized mechanism, dubbed $\mathcal{M}_{\mathtt{Q}}$, such that $\mathcal{M}_{\mathtt{Q}}$ satisfies differential privacy, cf. Theorem 1. To implement this, the Laplace mechanism adds – appropriately scaled – Laplace distributed noise to the aggregate. To determine this scale, we need to know (1) the desired privacy level ϵ and (2) the *global sensitivity* of the aggregation $\mathtt{Q}$. The latter quantifies the maximum difference of aggregates $\mathtt{Q}(D)$ and $\mathtt{Q}(D')$ provided any possibly existing neighboring databases D and D' (cf. Definition 4). Regarding data utility, small sensitivities are preferable. The global sensitivity depends on which definition of neighboring databases we use. For instance, according to Definition 4, the global sensitivity of the $\mathtt{Sum}$ aggregate is infinite, since $\Delta_{\mathtt{Sum}} = \alpha$ holds if we consider Definition 3.

Theorem 1 (Laplace Mechanism [8], [16])**.** *For a scalar aggregation query $Q : D \rightarrow \mathbb{R}$, the mechanism $\mathcal{M}_{\mathtt{Q}}$*

$$\mathcal{M}_{\mathtt{Q}} = \mathtt{Q} + Lap(\lambda)$$

that adds Laplacian noise with scale $\lambda = \frac{\Delta_{\mathtt{Q}}}{\epsilon}$ to the aggregate provides ϵ-differential privacy.

Definition 4 (Global Sensitivity [8])**.** *For a scalar aggregation query $\mathtt{Q} : D \rightarrow \mathbb{R}$, the global sensitivity is given by*

$$\Delta_{\mathtt{Q}} = \max_{\textit{neighbors } D, D'} d_{L_1}(\mathtt{Q}(D), \mathtt{Q}(D')).$$

3) Properties: Differential privacy features some useful properties that we divide in two categories. The first category are the privacy axioms, that every privacy framework should fulfill, in order to be meaningful [9]. The second category are composition theorems, which allow for mechanism design for outputting multiple aggregates, and are therefore the connecting bridge to differential privacy for continuous streams we discuss in the next section. We also prove them for Swellfish Privacy in the remainder. Note that here, we just review the properties shortly, and discuss them in detail there.

a) Privacy Axioms: There are privacy axioms that are commonly accepted and form "modern design guidelines" for privacy frameworks [9]. Differential privacy fulfills them, but, e.g., k-anonymity [17], another well-known privacy framework, does not [18]. The axioms are *post-processing immunity*, a.k.a. transformation in-variance, and *convexity*. Post-processing immunity states that arbitrary analysis over private aggregates do not compromise privacy. Convexity says that we can choose any privacy mechanism that fulfills the privacy definition, and that we can even randomize this choice.

b) Composition Properties: Further, when defining a privacy framework, one is always interested in composition properties [9]. That is, how privacy degenerates if we publish several aggregates with different privacy levels. Knowing composition properties helps for the privacy analysis of complex privacy mechanisms or algorithms. One differentiates between sequential and parallel composition [16]. The sequential composition theorem state that if we publish several aggregates over the *same* database, the privacy budgets sum up. The parallel composition theorem states that if we do the same over *disjunct* databases, the overall privacy level is the worst privacy budget.

C. w-Event Differential Privacy

While differential privacy is the current gold standard for static databases, w-event differential privacy is the same for streams. In the remainder, we state the definition of w-event differential privacy and appropriate sanitation mechanisms.

1) Definition: Definition 5 states the definition of w-event differential privacy. Initially, it is a straight-forward adaption of Definition 1 saying that a mechanism is w-event ϵ-differential private, if for all w-neighboring stream prefixes, the distributions of its output differ only by a factor of e^{ϵ}.

Definition 5 (w-Event ϵ-Differential Privacy [3])**.** *Let $\mathcal{M}$ be a randomized mechanism that takes as input a stream prefix of arbitrary size. We say that $\mathcal{M}$ satisfies w-*event ϵ-differential privacy *if for all $R \in Range(\mathcal{M})$, all w-neighboring stream prefixes $S_{\mathcal{T}}, S'_{\mathcal{T}}$, and all $\mathcal{T}$, holds that*

$$Pr[\mathcal{M}(S_{\mathcal{T}}) = R] \leq e^{\epsilon} \cdot Pr[\mathcal{M}(S'_{\mathcal{T}}) = R].$$

Particularly interesting is how w-neighboring stream prefixes are defined. This definition is given in Definition 6 and states that (1) for every point in time, D_t and D'_t are equal or neighboring (w.r.t. to Definition 2 or 3), and (2) all differing databases fit into a time window of at most w consecutive time stamps. That is, with w-event differential privacy, we protect events having at most length w.

Definition 6 (w-Neighboring Stream Prefixes [3]). *Let w be a positive integer, and $t, t_1, t_2 \leq \mathcal{T}$ three time stamps. Two stream prefixes $S_\mathcal{T}, S'_\mathcal{T}$ are* w-neighboring, *if*

1) for each D_t, D'_t with $D_t \neq D'_t$, it holds that D_t, D'_t are neighboring

2) for each $D_{t_1}, D_{t_2}, D'_{t_1}, D'_{t_2}$ with $t_1 < t_2$, $D_{t_1} \neq D'_{t_1}$ and $D_{t_2} \neq D'_{t_2}$, it holds that $t_2 - t_1 + 1 \leq w$.

2) w-Event Differential Private Mechanisms: Having defined w-event differential privacy, we aim at mechanisms $\mathcal{M}$ satisfying the definition. To this end, an important observation is Theorem 2, that is an equivalent to the sequential composition theorem of differential privacy. It states that we can apply a differentially private mechanism $\mathcal{M}_t$ to each static database D_t for every time stamp t, as long as we ensure that the sum of the budget of the $\mathcal{M}'_t s$ does not exceed ϵ for every rolling window of size w.

Theorem 2 (Composition [3]). *Let $\mathcal{M}$ be a mechanism processing a stream prefix $S_\mathcal{T} =<D_1, .., D_\mathcal{T}>$, and outputting a transcript $R =<r_1, .., r_\mathcal{T}>$. If we can decompose $\mathcal{M}$ into $\mathcal{T}$ sub-mechanisms $\mathcal{M}_1, .., \mathcal{M}_\mathcal{T}$, s.t. $\mathcal{M}_t(D_t) = r_t$, each $\mathcal{M}_t$ has independent randomness and achieves ϵ_t- differential privacy, than, $\mathcal{M}$ satisfies w-event differential privacy if*

$$\forall t \in [1, \mathcal{T}] : \sum_{k=t-w+1}^{t} \epsilon_k \leq \epsilon.$$

For instance, by leveraging Theorem 2, to ensure w-event ϵ-differential privacy, we can implement each sub-mechanism with the Laplace mechanism from the differential privacy framework, i.e., $\mathcal{M}_t = \mathcal{M}_\mathrm{Q}$, provided with a budget of $\frac{\epsilon}{w}$ each. That is, we split the privacy budget uniformly over the time stamps in the window. We call this naïve mechanism Uniform(w,ϵ,Δ_Q). However, other mechanisms exists, that distribute the budget in an adaptive way. A current state-of-the-art mechanism is RescueDP [5] that is known to publish `Count` and `Histogram` aggregates with good data utility. We will use both mechanisms as competitors in our evaluation in the remainder of this paper.

III. Swellfish Privacy

In this section, we introduce our privacy framework Swellfish Privacy, that is a generalization of the w-event differential privacy framework. The core idea is to allow individuals to specify temporally varying privacy specifications, such that we have to protect (1) long and/or (2) high-power events *only* in times, in that they have to be protected, ultimately resulting in high data utility.

In this section, we follow – as far as possible – the same structure as Section II-B where we introduced differential privacy. We first define Swellfish Privacy and give a glance of it. This definition is based on the notation of *privacy specifications* and stream prefixes that are *neighboring with respect to such a specification*. Therefore, after stating the definition, we introduce our notation of privacy specifications, and our definition of neighboring stream prefixes. In the fourth part of this section, we then address fundamental properties of our framework, like the fulfillment of the privacy axioms proving. Last, we outline the relationship of Swellfish Privacy to other privacy frameworks from literature, revealing further insights in the subtleties of Swellfish Privacy.

A. Definition of Swellfish Privacy

In this section, we define Swellfish Privacy and give an intuition for it, before we go into details in the following sections. For didactic reasons, we first introduce our definition of Swellfish Privacy for one privacy specification. However, in real applications, one privacy specification refers to the privacy needs of one individual. As generally, every individual has some privacy needs, we afterwards extend our definition allowing to protect multiple privacy specifications.

1) Swellfish Privacy at a Glance: Definition 7 states our definition of Swellfish Privacy for one privacy specification.

Definition 7 ($\mathcal{P}$-Swellfish Privacy). *Let $\mathcal{M}$ be a randomized mechanism that processes a stream prefix $S_\mathcal{T}$ of arbitrary size, and $\mathcal{P}$ a privacy specification. The mechanism $\mathcal{M}$ gives $\mathcal{P}$-Swellfish Privacy, iff for all stream prefixes $S_\mathcal{T}, S'_\mathcal{T}$ that are neighboring w.r.t. $\mathcal{P}$, short $(S_\mathcal{T}, S'_\mathcal{T}) \in \mathcal{N}(\mathcal{P})$, all secrets $s \in \mathcal{P}$, and all $R \in Range(\mathcal{M})$, holds*

$$Pr[\mathcal{M}(S_{s.\mathcal{J}}) = R] \leq e^{s.\epsilon} \cdot Pr[\mathcal{M}(S'_{s.\mathcal{J}}) = R].$$

As w-event differential privacy, Swellfish Privacy relies on the concept of indistinguishability of aggregates for neighboring stream prefixes. However, the core difference between them is that our definition of neighboring stream prefixes is based on *privacy specifications*. A privacy specification $\mathcal{P}$ consists of several secrets that should be hidden. A secret $s \in \mathcal{P}$ consists of three components:

Event Every secret refers to an event $s.\mathcal{E}$ in a house, e.g., cooking, an individual assesses as privacy critical. Different events might have *different length* T and *height of power consumption* p that has to be hidden in the aggregate.

Hiding Interval Every secret states a *hiding interval* $s.\mathcal{J}$, as events are worth to be hidden at different points in time. For instance, an individual might aim at hiding a cooking event in the night, but not at lunch time, as it is common to cook at lunch time, but it is not in the night. As consequence, in Definition 7, we only claim for indistinguishability during the hiding intervals.

Privacy Level As described, it is generally impossible to provide perfect indistinguishability of aggregates. Therefore, every secret is equipped with a secret-specific privacy level

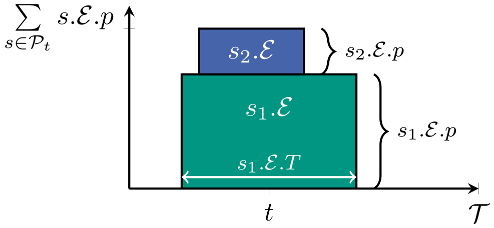

Fig. 1. Two events, both relevant at time t.

$s.\epsilon$. With that, we account for the fact that some secrets might have to be hidden more accurately than others.

In contrast, w-event differential privacy presumes that we have to hide (1) the event with longest time duration and power consumption during (2) the entire lifetime of the stream with (3) the same, and therefore relatively small, privacy level. Secrets should be hidden *simultaneously* with all other secrets contained in a specification and may have overlapping hiding intervals, cf. Figure 1.

2) Swellfish Privacy for Multiple Specifications: If multiple privacy specifications are given, e.g., one per individual in a city, we say that a mechanism provides Swellfish Privacy, iff it provides Swellfish Privacy for any of these, cf. Definition 8. This is the naturally way to define it. As we see in the remainder, if we have a mechanism that provides Swellfish Privacy for one specification, we can extend it to a mechanism for multiple specifications easily. Therefore, we focus on the above definition for one specification initially.

Definition 8 (($\mathcal{P}_1, \ldots, \mathcal{P}_n$)-Swellfish Privacy)**.** *Let $\mathcal{M}$ be a randomized mechanism that processes a stream prefix $S_\mathcal{T}$ of arbitrary size, and $\mathcal{P}_1, \ldots, \mathcal{P}_n$ privacy specifications. $\mathcal{M}$ provides $(\mathcal{P}_1, \ldots, \mathcal{P}_n)$-Swellfish Privacy, iff $\mathcal{M}$ provides $\mathcal{P}_i$-Swellfish Privacy for all $\mathcal{P}_i$, $1 \leq i \leq n$.*

B. Declarative Privacy Specifications

In this section, we state how individuals can express their privacy specifications. The privacy specifications are *declarative*, meaning individuals can just specify their needs, and do not have to think about the implementation. Thus, in the following, we initially only elaborate on how the specifications can be expressed by the individuals, before we state how our framework implements them via a mechanism in Section IV. A privacy specification consists of several secrets that should be hidden in the aggregate simultaneously. Thus, in this section, we first discuss how to express secrets. Then, we introduce our notation of declarative privacy specifications.

1) Secrets: A literature search reveals that generally, an individual aims at hiding certain events that correspond to activities that he performs at home, and might be visible when inspecting the aggregated power consumption, but are limited in time. Respective examples are occupancy [19] for two weeks or the usage of an appliance [20]. Therefore, we first introduce what we mean by an event.

a) Events: Definition 9 states the definition of an event. An example of an event is washing clothes with the washing machine. If we consider a washing machine that consumes $p = 1$ kW, and washing clothes lasts $T = 8 \cdot \tau$ minutes, we model this by the event $\mathcal{E} = (1, 8)$. In reality, the power consumption of events might be different for different time stamps during the duration of an event. For instance, a washing machine might consume less power in the beginning, than in the end while spinning. We assume that p is the maximum consumption of the event, as this is the value we actually have to hide.

Definition 9 (Event)**.** *An event $\mathcal{E} = (p, T)$ is a tuple consisting of a* power value *p in kW reflecting the maximum consumption of the event, and a* number of time stamps *T reflecting the temporal duration of the event.*

b) From Events to Secrets: Having defined events, we now state how to define a secret based on events with Definition 10.

Definition 10 (Secret)**.** *A secret is a tuple $s = (\mathcal{E}, \mathcal{J}, \epsilon)$, where $\mathcal{E}$ is an* event*, $\mathcal{J} = [t_B, t_E]$ is a* hiding interval *with length $|\mathcal{J}| \geq \mathcal{E}.T$, and $\epsilon > 0$ a desired* privacy level*. $\mathcal{J}$ is the time interval in which the event $\mathcal{E}$ should be hidden. Further, ϵ is the desired privacy level.*

A secret consists, apart of an event, of a hiding interval $\mathcal{J}$ and a privacy level ϵ. The rational behind the hiding interval is that generally, one does not aim hiding every event during all the time, but in different time intervals. In the remainder, we say that a secret is *relevant* at time t, if $t \in \mathcal{J}$. See Example 1 for an example of a secret.

Example 1 (Secret)**.** *Assume we want to hide a $T = 2$ long usage of the oven ($p = 3$ kW) with privacy level $\epsilon = 1$ between time stamps, say, 1 and 4, reflecting lunch time on Monday, the 4th of February. This can be formalized by a secret as follows: Event $\mathcal{E} = (3, 2)$, and Secret $s = (\mathcal{E}, \mathcal{J} = [1, 4], \epsilon = 1)$.*

We want to compare secrets regarding their conservativity. Generally, we say that a secret is *conservative*, if its associated event (1) is long, (2) has a high consumption, and (3) the secret has a small privacy level. This is a multi-criteria evaluation, and it is therefore not possible to put this into a quantitative metric, with which we can compare any pair of secrets. However, given a set of secrets, we can say that a secret is dominant iff it is more conservative than any other secret in the set concerning all three criteria.

2) Privacy Specification: A privacy specification $\mathcal{P}$ is a set of secrets. Definition 7 reveals that neighboring stream prefixes are defined w.r.t. a privacy specifications. This means, that all secrets contained should be implemented simultaneously. This is in particular challenging in case the hiding intervals of the secrets overlap, as we see in the remainder. Like for secrets, we aim at comparing privacy specifications regarding their conservativity. We say that a specification $\mathcal{P}_1$ dominates another specification $\mathcal{P}_2$ at time t, if the respective dominant secrets of $\mathcal{P}_{1,t}$ and $\mathcal{P}_{2,t}$ relevant at time t dominate each other.

$S_{\mathcal{T}}$ =

Individual	$t=1$	$t=2$	$t=3$	$t=4$
$i=1$	0.2	0.3	0.2	3.1
$i=2$	0.4	0.4	0.3	3.2

$S'_{\mathcal{T}}$ =

Individual	$t=1$	$t=2$	$t=3$	$t=4$
$i=1$	0.2	3.3	3.2	3.1
$i=2$	0.4	0.4	0.3	3.2

Fig. 2. Neighboring stream prefixes w.r.t. $\mathcal{P}$.

Definition 11 (Privacy Specification). *A privacy specification $\mathcal{P}$ is a set of secrets that should be hidden simultaneously.*

C. Neighboring Databases and Stream Prefixes

The definition of w-event differential privacy states that aggregates should be indistinguishable for neighboring stream prefixes. Thereby, stream prefixes are neighbors, only if *all static* databases D_t differ in one individual. In contrast, our definition of neighboring stream prefixes is based on a privacy specification. Thus, we first introduce neighboring *static* databases with respect to a privacy specification $\mathcal{P}$. Then, we extend it to stream prefixes.

1) Neighboring Static Databases w.r.t. $\mathcal{P}$: Definition 12 states the definition of neighboring static databases with respect to a privacy specification. It builds up on the definition of α-L_1-neighboring databases, but bounds explicitly the required α, i.e., the L_1-distance between the databases. The rational behind α is a worst-case treatment. I.e., we ask which is the maximum consumption difference we have to hide at time t according to the given privacy specification. Therefore, we consider all secrets that are relevant at time t, and sum up the consumption of the corresponding event $\mathcal{E}$, cf. Figure 1.

Definition 12 (Neighboring Databases w.r.t. $\mathcal{P}$). *Let $\mathcal{P}$ be a privacy specification and $\mathcal{P}_t = \{s \in \mathcal{P} \mid t \in s.\mathcal{J}\}$ the secrets relevant at time t. Two databases D_t, D'_t are neighboring w.r.t. a privacy specification $\mathcal{P}$, short $(D_t, D'_t) \in \mathcal{N}(\mathcal{P})$, if $(D_t, D'_t) \in \mathcal{N}_1(\sum\limits_{s \in \mathcal{P}_t} s.\mathcal{E}.p)$, i.e.,*

$$d_{L_1}(D_t, D'_t) = \sum_{s \in \mathcal{P}_t} s.\mathcal{E}.p.$$

2) Neighboring Stream Prefixes w.r.t. $\mathcal{P}$: Now, Definition 13 introduces neighboring stream prefixes with respect to a privacy specification. The intuition behind the definition is that neighboring streams differ by at most one cycle of every event $\mathcal{E}$ in its respective hiding interval. Figure 2 illustrates the definition by means of a privacy specification that consists only of the secret introduced in Example 1.

Definition 13 (Neighboring Stream Prefixes w.r.t. $\mathcal{P}$). *Two stream prefixes $S_{\mathcal{T}}, S'_{\mathcal{T}}$ are neighboring w.r.t. a privacy specification $\mathcal{P}$, short $(S_{\mathcal{T}}, S'_{\mathcal{T}}) \in \mathcal{N}(\mathcal{P})$, if:*

1) for every $t \in [0, \mathcal{T}]$, $D_t, D'_t \in \mathcal{N}(\mathcal{P})$

2) $S_{\mathcal{T}}$ differs from $S'_{\mathcal{T}}$ by one occurrence of every event associated to a secret in the respective hiding interval.

D. Properties

As discussed in Section II-B3, every meaningful privacy framework should feature the privacy axioms. Further, composition properties are desired. We now formulate and address these axioms and properties for Swellfish Privacy.

1) Privacy Axioms: As stated in Section II-B3, we have to prove that Swellfish Privacy features post-processing immunity and convexity. We begin the the convexity axiom, as it is the key to proof the other theorems. Convexity says that all privacy mechanisms that satisfy Swellfish Privacy are of equal value in the sense that they produce outputs that sanitize the input streams sufficiently.

Theorem 3 (Convexity). *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two mechanisms that fulfill $\mathcal{P}$-Swellfish Privacy and have independent randomness, and $p \in [0,1]$. Let $\mathcal{M}_p$ the algorithm that runs $\mathcal{M}_1$ with probability p, and $\mathcal{M}_2$ with probability $1-p$, where p is independent of the data and the randomness in $\mathcal{M}_1$ and $\mathcal{M}_2$. Then, $\mathcal{M}_p$ fulfills $\mathcal{P}$-Swellfish Privacy.*

Proof: Let s be a secret. Than it holds

$$\begin{aligned}&\frac{Pr[\mathcal{M}_p(S_{s.\mathcal{J}}) \in R]}{Pr[\mathcal{M}_p(S'_{s.\mathcal{J}}) = R]}\\ =&\frac{p \cdot Pr[\mathcal{M}_1(S_{\{s.\mathcal{J}\}}) = R] + (1-p) \cdot Pr[\mathcal{M}_2(S_{s.\mathcal{J}}) \in R]}{p \cdot Pr[\mathcal{M}_1(S'_{s.\mathcal{J}}) = R] + (1-p) \cdot Pr[\mathcal{M}_2(S'_{s.\mathcal{J}}) = R]}\\ \leq& p \cdot e^{s.\epsilon} + (1-p) \cdot e^{s.\epsilon} = e^{s.\epsilon}.\end{aligned}$$

□

Theorem 4 (Post-Processing Immunity). *Let $\mathcal{M}$ be a privacy mechanism that fulfills $\mathcal{P}$-Swellfish Privacy, and let A be a randomized algorithm whose input space contains the output space of $\mathcal{M}$ and whose randomness is independent of both the data and the randomness in $\mathcal{M}$. Then $A \circ \mathcal{M}$ also satisfies $\mathcal{P}$-Swellfish Privacy.*

Proof: Analogous to the proof in [7] for differential privacy, by using convexity.

2) Composition Properties: As stated in Section II-B3, we are interested in how the privacy in the Swellfish framework degenerates under sequential and parallel composition. To this end, Theorem 5 states that under sequential composition, the privacy levels for every secret sum up. This is similar to differential privacy. Further, Theorem 6 states that under parallel composition, where we split the stream in two disjoint parts, privacy level degenerates for secrets that are relevant in both parts only.

Theorem 5 (Sequential Composition). *Let $S_{\mathcal{T}}$ be a stream prefix, and $\mathcal{M}_1$, $\mathcal{M}_2$ be two mechanisms that provide $\mathcal{P}$-Swellfish Privacy. Then, the mechanism $\mathcal{M} = (\mathcal{M}1, \mathcal{M}_2)$ that outputs $(\mathcal{M}_1(S_{\mathcal{T}}), \mathcal{M}_2(S_{\mathcal{T}})$ satisfies $\mathcal{P}$-Swellfish Privacy, but the privacy levels degenerate to $2 \cdot s.\epsilon$ for each secret s.*

Proof: Let s be a secret. Than it holds

$$\frac{Pr[\mathcal{M}(S_{s.\mathcal{J}}) = R]}{Pr[\mathcal{M}(S'_{s.\mathcal{J}}) = R]} = \sum_{i=1,2} \frac{Pr[\mathcal{M}_i(S_{s.\mathcal{J}}) = R]}{Pr[\mathcal{M}_i(S'_{s.\mathcal{J}}) = R]}$$
$$\leq \sum_{i=1,2} e^{s.\epsilon} = e^{2\cdot\epsilon}. \qquad \square$$

Theorem 6 (Parallel Composition)**.** *Let $D_{\mathcal{T}}$ be a stream prefix, and $\mathcal{M}_1$, $\mathcal{M}_2$ be two mechanisms that provide $\mathcal{P}$-Swellfish Privacy. Divide $S_{\mathcal{T}} = <D_1, .., D_{\mathcal{T}}>$ in $S_{[1,T_1]} = < D_1, .., D_{T_1} >$ and $S_{(T_1,\mathcal{T}]} = <D_{T_1+1}, .., D_{\mathcal{T}}>$. Then, $S_{[1,T_1]}$ and $S_{(T_1,\mathcal{T}]}$ are itself stream prefixes. Further, divide $\mathcal{P}$ in the three disjoint subsets*

- $\mathcal{P}_{[1,T_1]} = \{s$ *is only relevant in* $[1, T_1]\}$,
- $\mathcal{P}_{(T_1,\mathcal{T}]} = \{s$ *is only relevant in* $(T_1, \mathcal{T}]\}$ *and*
- $\mathcal{P}_{\cap} = \{s$ *is relevant in* $[1, T_1]$ *and* $(T_1, \mathcal{T}]\}$.

Now, consider the mechanism $\mathcal{M} = (\mathcal{M}_1, \mathcal{M}_2)$ that outputs both sub-streams $(\mathcal{M}_1(D_{[0,T_1]}), \mathcal{M}_2(D_{(T_1,\mathcal{T}]})$.

1) *If $\mathcal{P}_{\cap} = \emptyset$, κ satisfies $\mathcal{P}$-Swellfish Privacy (with* no *degeneration of the privacy level).*
2) *Otherwise, the mechanism $\mathcal{M}$ satisfies $\mathcal{P}$-Swellfish Privacy, but the privacy levels degenerate to $2 \cdot s.\epsilon$ for each secret $s \in \mathcal{P}_{\cap}$.*

Proof: Case (1) results directly from the fact that both sub-streams do not share secrets. The proof of case (2) is similar to Theorem 5. □

E. Relationships to other Privacy Frameworks

To provide a better understanding of Swellfish Privacy, we now differentiate it to existing privacy frameworks from literature. First, we focus on the Pufferfish [9] and Blowfish framework [10], as they are general, and many existing privacy frameworks are instances of one of them. Second, we look at other frameworks not being instances of them, including the ones that are special for power consumption data.

1) Pufferfish and Blowfish Privacy: The Pufferfish [9] and Blowfish framework [10] are generalizations of differential privacy. In both frameworks, it is possible (1) to specify secrets that should be hidden, and (2) define possible data generations processes to restrict adversarial background knowledge. Both frameworks differ in how (2) can be specified. Swellfish Privacy, like differential privacy, does not restrict the possible data generation processes. In other words, it presumes a functional dependence between databases at different points in time, that are known to the adversary. As both frameworks allow for the specification of an unrestricted set of data generation processes, this part of Swellfish Privacy could be modeled with the frameworks. However, this is not the case for the privacy specifications. While it would be possible to formulate the simultaneous hiding of activities in terms of Pufferfish/Blowfish secrets by building bounding boxes, it is not possible to model the secret-dependent privacy level. Therefore, Swellfish Privacy is not an instance of the Pufferfish/Blowfish and demonstrates their borders.

2) Others: Personalized differential privacy [11] offers individual privacy levels. However, it does not allow for privacy specifications, and the streaming setting where individual privacy level preferences may change over time. INPACT [21] and PrivEnergy [22] allow for individual privacy specifications that are different from ours, in particular, more abstract, but are not suitable for the streaming setting, as their mechanism presume that we know the stream prefixes completely.

IV. MECHANISM DESIGN

After defining Swellfish Privacy, we now aim at finding a mechanism that satisfies the definition. In this section, we design mechanisms based on Laplace perturbation. Thus, as usual, we need a notation of sensitivity. Therefore, we first introduce our notation of temporal sensitivity. Next, we introduce the first mechanisms, which form the baseline we improve afterwards. Baseline mechanisms are all w-event mechanisms if parametrized correctly. This proves also that Swellfish Privacy is a generalization of w-event differential privacy. However, the baseline mechanisms do not exploit the temporal variability of privacy specifications. Therefore, afterwards, we design the Swellfish mechanism exploiting them. Finally, we compare the utility of all mechanisms from a theoretic perspective, before we compare them empirically in the remainder.

A. Temporal Sensitivity

Definition 14 states our notation of temporal sensitivity. It is a straight-forward adaption of the global sensitivity to our setting where relevant secrets, and therefore also the sensitivity, change over time. Observe that, for `Sum` aggregation, the temporal sensitivity is

$$\Delta^t_{\texttt{Sum}} = \sum_{s \in \mathcal{P}_t} s.\mathcal{E}.p.$$

The rational is that at a given time stamp t, all events that are relevant at time t can happen simultaneously. Therefore, we have to sum up their power.

Definition 14 (Temporal Sensitivity)**.** *For an aggregation query $\mathcal{Q} : D \to \mathbb{R}$, the temporal sensitivity at time t is*

$$\Delta^t_{\mathcal{Q}} = \max_{(D_t, D'_t) \in \mathcal{N}(\mathcal{P})} d_{L_1}(\mathcal{Q}(D_t), \mathcal{Q}(D'_t)).$$

B. Baseline Mechanisms

We now design our baseline mechanism, which is any w-event mechanism parametrized accordingly. For the baseline mechanism, as well as for the Swellfish mechanism later on, we say how to design the mechanism for one, as well as multiple privacy specifications, as the latter is frequently needed in real applications as already discussed.

1) Baseline Mechanism for One Privacy Specification: As baseline, we use any w-event differentially private mechanism. As Theorem 7 shows, every such mechanism provides Swellfish Privacy, if we parametrize the mechanism accordingly. The rationale behind the parametrization can be intuitively explained with the concept of dominant secrets introduced in

Section III-B. In Theorem 7, we build a secret that dominates all secrets in $\mathcal{P}$, and makes sure that we also account for simultaneous occurring events *anywhere in the stream* by using the maximum of the temporal sensitivity, as discussed above.

Theorem 7. *Let $\mathcal{M}$ be a mechanism that implements w-event ϵ-differential privacy for*

- *Window size $w = \max_{s \in \mathcal{P}} s.\mathcal{E}.T$,*
- *Privacy level $\epsilon = \min_{s \in \mathcal{P}} s.\epsilon$ and*
- *Sensitivity $\Delta_{\mathcal{Q}} = \max_{1 \leq t \leq T} \Delta^t_{\mathcal{Q}}$.*

*Then, it fulfills $\mathcal{P}$-*Swellfish *Privacy.*

Proof: Let s be a secret, $\mathcal{M}$ an w-event differential privacy mechanism implemented with the parameters stated above, and w.l.o.g. $[1,T] \subseteq \mathcal{J}$ and interval consisting of T consecutive time stamps. It holds that $T \leq w$. Therefore,

$$\begin{aligned} &\frac{Pr[\mathcal{M}(S_{[1,T]}) = R]}{Pr[\mathcal{M}(S'_{[1,T]}) = R]} \\ &\leq e^{\epsilon} && \mathcal{M} \text{ provides } w\text{-event DP} \\ &\leq e^{s.\epsilon}. && \text{choice of } \epsilon \quad \square \end{aligned}$$

2) Baseline Mechanism for Multiple Privacy Specifications: If we want to implement multiple privacy specifications, we have to calculate the parameter values as in Theorem 7 for the dominant specification. That is, for every specification, we determine the parameters according to Theorem 7 and, then in turn, take the maximum of them. The proof is straight-forward.

As we see, it is possible to implement privacy specifications with w-event mechanisms, but these mechanism do not exploit the hiding intervals, as they use the same parameters during the whole life-time of the stream. For instance, they provide w-event differential privacy for the longest event specified in $\mathcal{P}$, even the the corresponding secret is relevant for a short time only. Therefore, in the following, we state a mechanism that exploits this.

C. Swellfish *Mechanism*

To exploit the hiding intervals, we propose a mechanism that adds independent noise with a time-specific scale λ_t to the aggregate. That is, in contrast to the baseline mechanism, where we use the same parameter settings during the entire lifetime of the stream, the noise scale is different for different points in time. For implementation, we rely on the Laplace mechanism.

Theorem 8 states our Swellfish mechanism implementing *one* privacy specification. The rational behind is similar to the one of the baseline mechanism, with the important difference that (1) we can use the temporal sensitivity directly, and have not to maximize over it, and (2) build the dominant secret only over secrets *actually relevant* at time t.

Theorem 8 (Swellfish($\mathcal{P}$))**.** *Let $\mathcal{Q}$ be an aggregation query, and $\mathcal{P}_t = \{s \in \mathcal{P} \mid t \in \mathcal{J}\}$ be the set of secrets relevant at time t. Then, the Laplace mechanism $\mathcal{M}_t$ provides $\mathcal{P}$-*Swellfish *Privacy if it adds at time t independent noise to the aggregate having scale*

$$\lambda^{\mathcal{P}}_t = \Delta^t_{\mathcal{Q}} \cdot \frac{\max_{s \in \mathcal{P}_t} s.\mathcal{E}.T}{\min_{s \in \mathcal{P}_t} s.\epsilon}.$$

Proof: Let s be a secret, and w.l.o.g. $[1,T] \subseteq s.\mathcal{J}$ any subinterval of $s.\mathcal{J}$. First, we make the following two observations, we can put together to proof the theorem afterwards:

First observation: It holds

$$\begin{aligned} Pr[\mathcal{M}_t(D_t) = S] =& Pr[\text{Lap}(\lambda^{\mathcal{P}}_t) = S - \mathcal{Q}(D_t)] \\ =& \frac{1}{2\lambda^{\mathcal{P}}_t} e^{-\frac{|S - \mathcal{Q}(D_t)|}{\lambda^{\mathcal{P}}_t}} \end{aligned}$$

by the definition of the Laplace distribution. Further, it holds

$$\frac{|\mathcal{Q}(D'_t) - \mathcal{Q}(D_t)|}{\lambda^{\mathcal{P}}_t} = \frac{|\mathcal{Q}(D'_t) - \mathcal{Q}(D_t)| \cdot s.\epsilon}{\Delta_s \mathcal{Q} \cdot T} \leq \frac{e^{s.\epsilon}}{T}.$$

Putting things together, we have

$$\begin{aligned} \ln|\frac{Pr[\mathcal{M}_t(D_t) = S]}{Pr[\kappa(D'_t) \in S]}| &= -\frac{|S - \mathcal{Q}(D_t)|}{\lambda} + \frac{|S - \mathcal{Q}(D'_t)|}{\lambda_t} \\ &\leq \frac{|\mathcal{Q}(D'_t) - \mathcal{Q}(D_t)|}{\lambda_t} = \frac{|\mathcal{Q}(D'_t) - \mathcal{Q}(D_t)|}{\lambda_t} \\ &\leq \frac{e^{s.\epsilon}}{T}. \end{aligned}$$

Second observation: As all $\mathcal{M}_t$ are independent, it holds that

$$\begin{aligned} Pr[(\mathcal{M}_1(D_1), .., \mathcal{M}_T(D_T)) = (S_1, .., S_T)] \\ = \prod_{1 \leq t \leq T} Pr[\mathcal{M}_t(D_t) = S_t] \end{aligned}$$

Putted together, we obtain

$$\begin{aligned} &\frac{Pr[(\mathcal{M}_1(D_1), .., \mathcal{M}_T(D_T)) = (R_1, .., R_T¿]}{Pr[(\mathcal{M}_1(D_1), .., \mathcal{M}T(D_T))] = (R_1, .., R_T)]} \\ =& \frac{\prod_{1 \leq t \leq T} Pr[\mathcal{M}_t(D_t) = R_t]}{\prod_{1 \leq t \leq T} Pr[\mathcal{M}_t(D'_t) = R_t]} && \text{by Obs. 2} \\ =& \prod_{1 \leq t \leq T} \frac{Pr[\mathcal{M}_t(D_t) = R_t]}{Pr[\mathcal{M}_t(D'_t) = R_t]} \\ \leq& \prod_{1 \leq t \leq T} e^{\frac{s.\epsilon}{T}} && \text{by Obs. 1} \\ =& e^{s.\epsilon} \quad \square. \end{aligned}$$

Next, Theorem 9 states how to implement *multiple* specifications. To this end, we have to use the maximum scale at any time t considering all privacy specifications given. This means that the we build the dominant privacy specification, as for the baseline mechanism. However, again, it is important that in contrast to the baseline mechanism, we only have to do this for secrets *actually relevant* at time t.

Theorem 9 (Swellfish($\mathcal{P}_1, .., \mathcal{P}_n$))**.** *Let $\{\mathcal{P}_1, ..., \mathcal{P}_i, ..., \mathcal{P}_n\}$ be a set of privacy specifications. Then, the Laplace mechanism $\mathcal{M}_t$ provides $\mathcal{P}_i$-*Swellfish *Privacy for every $\mathcal{P}_i$ if it adds at time t independent noise to the aggregate having scale*

$$\lambda^{\cup}_t = \max \lambda^{\mathcal{P}_i}_t.$$

Proof: Analogous to the proof of Theorem 8. □

D. Theoretic Utility Analysis

As usual, we are interested in mechanisms that provide good data quality. Usually, we measure data utility with the relative error in the aggregate we can expect. So far, we proposed baseline mechanisms and Swellfish mechanism that both satisfy Swellfish Privacy. We argue that the data utility we can expect from the Swellfish mechanism is generally better, as follows: At every point in time t, our mechanism adds Laplace distributed noise with scale $\lambda_t^{\mathcal{P}}$ to the aggregate. The smaller the scale is, the lesser the Laplace distribution spreads out around its mean of zero. Therefore, the smaller the values for the scale are, the better data utility can we expect. As $\Delta_{\mathrm{Q}}^t \leq \max_{1 \leq t \leq \mathcal{T}} \Delta_{\mathrm{Q}}^t$, $\max_{s \in \mathcal{P}_t} s.\mathcal{E}.T \leq \max_{1 \leq t \leq s.\mathcal{E}.T} s.\mathcal{E}.T$ and $\min_{s \in \mathcal{P}_t} s.\epsilon \geq \min_{1 \leq t \leq \mathcal{T}} s.\epsilon$, the scale of Swellfish mechanism is smaller or equal the scale of the baseline Uniform mechanism. Therefore, we expect better data utility from Swellfish mechanism than from the one of the baseline mechanism. We verify this via experiments later on in this paper.

V. Approximate Swellfish Privacy

In the last two sections, we proposed Swellfish Privacy and a mechanism that satisfies it. However, the noise scale of the Swellfish mechanism is in particular proportional to the length T of the dominant event, which might be high. However, literature indicates that we can reduce this factor to $\sqrt{T}$ by using *approximate* mechanisms. Therefore, in this section, we propose approximate Swellfish Privacy. It is inspired by (ϵ, δ)-differential privacy, that is also known as *approximate differential privacy*. In this section, we first recall approximate differential privacy and its composition properties, showing why we can reduce the factor to $\sqrt{T}$. Then, we define approximate Swellfish Privacy, propose a mechanism that satisfies it and compare it to pure Swellfish Privacy.

A. Preface – (ϵ, δ)-Differential Privacy

In this section, we first recall the definition of (ϵ, δ)-differential privacy and a corresponding Laplace mechanism for stream prefixes.

1) Definition: Definition 15 states the definition of approximate differential privacy. It states that we allow for a probability of approximately $\delta \geq 0$ that the stream prefixes are indistinguishable more than by the factor $e^{s.\epsilon}$. For $\delta = 0$, approximate differential privacy is same as pure differential privacy, i.e., Definition 1. For every $\delta > 0$, the privacy guarantee is weaker than pure differential privacy. Thus, the smaller δ is, the better is the privacy provided.

Definition 15 ((ϵ, δ)-Differential Privacy [23])**.** *Let $\epsilon, \delta \geq 0$. A randomized mechanism $\mathcal{M}$ gives (ϵ, δ)-differential privacy if for all neighboring databases D and D', and all $R \in$ Range($\mathcal{M}$),*

$$Pr[\mathcal{M}(D) = R] \leq e^{\epsilon} \cdot Pr[\mathcal{M}(D') = R] + \delta.$$

Approximate differential privacy is controversial discussed, as much care must be taken when choosing δ. For instance, if $\delta = \frac{1}{|D|}$, a function that publishes complete true values of a subset of individuals in D is differential private [7]. This is never the case for pure differential privacy, regardless of the chosen ϵ. To avoid such an unexpected privacy loss, δ should be negligible in the size of the database [7].

To define approximate differential privacy for streams, we just extend the definition of w-event differential privacy, i.e., Definition 5, with parameter δ in the same way as in Definition 15. We dub the obtained definition *w-event (ϵ, δ)-differential privacy*.

2) Laplace Mechanism: Regarding mechanism design for data steams, approximate differential privacy has a big advantage towards pure differential privacy: it behaves much more nicely under sequential composition as follows. Let $\mathcal{M}$ be a ϵ-differential private mechanism for static databases. According to Theorem 2, applied to streams for every point in time separately, the privacy level sums up over time. Therefore, e.g., within the Uniform mechanism, we provide each $\mathcal{M}$ with budget $\frac{w}{\epsilon}$, to achieve privacy level ϵ in total, as explained in Section II-C2. This bound is tight for pure differential privacy [14]. However, it can be tightened for approximate differential privacy, as explained in [14]. As a consequence, we obtain the Laplace mechanism stated in Theorem 10 for approximate differential privacy. Recall that the scale for the Uniform mechanism for pure differential privacy is $\lambda = \Delta_{\mathrm{Q}} \cdot \frac{w}{\epsilon}$, which is much larger than the scale stated in Theorem 10.

Theorem 10 (Uniform(w,ϵ,δ,Δ_{Q}))**.** *Let $\epsilon > 0$, $\delta \in (0, 1]$, $S_{\mathcal{T}}$ be a stream prefix and $\mathcal{M}$ the Laplace mechanism that adds Laplacian noise to the aggregate having scale*

$$\lambda = \Delta_{Q} \cdot \frac{2}{\epsilon} \cdot \sqrt{w \cdot ln(e + \frac{\epsilon}{\delta})}.$$

Then, $\mathcal{M}$ provides w-event (ϵ,δ)-differential privacy.

Proof: Follows from [14] by using the relation $v = \frac{1}{2}\lambda^2$ between variance v and scale λ of the Laplace distribution. □

B. $(\mathcal{P}, \delta)$- Swellfish Privacy

Analogous to approximate differential privacy, we are able to define approximate Swellfish Privacy. In the following, we state this definition and a corresponding mechanism.

1) Definition: Definition 16 states the definition of approximate Swellfish Privacy. Observe that δ is global, i.e., the same for every secret, as we intend to fix δ to a value, and do not leave the choice of δ to the individuals. The rational is to avoid unexpected privacy loss due to non-appropriate choices of δ like illustrated above.

Definition 16 (($\mathcal{P}, \delta$)-Swellfish Privacy)**.** *Let $\delta > 0$, $\mathcal{M}$ be a randomized mechanism that takes as input a stream prefix $S_{\mathcal{T}}$ of arbitrary size, $\mathcal{P}$ a privacy specification. The function $\mathcal{M}$ gives $(\mathcal{P}, \delta)$-Swellfish Privacy, iff for all neighboring stream prefixes $(S_{\mathcal{T}}, S'_{\mathcal{T}}) \in \mathcal{N}(\mathcal{P})$, all secrets $s \in \mathcal{P}$, and all $R \in$ Range($\mathcal{M}$), holds*

$$Pr[\mathcal{M}(S_{s.\mathcal{J}}) = R] \leq Pr[\mathcal{M}(S'_{s.\mathcal{J}}) = R] \cdot e^{s.\epsilon} + \delta.$$

2) Swellfish Mechanism: Now, Theorem 11 states the corresponding mechanism that satisfies approximate Swellfish Privacy. If we compare it with Theorem 8 which states the Laplace mechanism for "pure“ Swellfish Privacy, we observe that the length T of an event enters into the noise scale only with a factor $\sqrt{T}$, where it enters with a linear factor within the noise scale of "pure“ mechanism. Therefore, in particular in the presence of long events, approximate Swellfish Privacy could have better data utility than "pure“ Swellfish Privacy. We verify this with experiments later on.

Theorem 11 (Swellfish($\mathcal{P}$,δ))**.** *Let $\delta \in (0,1]$, $\mathcal{Q}$ be an aggregation query, and $\mathcal{P}_t = \{s \in \mathcal{P} \mid t \in \mathcal{J}\}$ be the set of secrets relevant at time t. Then, the Laplace mechanism $\mathcal{M}_t$ that adds at time t independent noise with scale*

$$\lambda_t^{\mathcal{P}} = \Delta_{\mathcal{Q}} \cdot \max_{s \in \mathcal{P}_t} \frac{2}{s.\epsilon} \cdot \sqrt{s.\mathcal{E}.T \cdot ln(e + \frac{s.\epsilon}{\delta})}$$

to the aggregate provides ($\mathcal{P}$,δ)-Swellfish Privacy.

Proof: Analogous to the proof of Theorem 8 by using Theorem 10. □

VI. Experimental Evaluation

In this section, we describe and discuss the experimental evaluation of Swellfish Privacy. The objective is to evaluate the data utility of our mechanism in our main use-case, which is the continuous publishing of private `Sum` aggregates of power consumption streams. We do this in a realistic setting by means of two experiments having different investigation purposes. This is very challenging due to, e.g., the lack of realistic privacy specifications. Therefore, we first describe our methodology, including a detailed description how we address the challenges in our evaluation. Then, we state and discuss the results of the experiments.

A. Methodology

In this section, we describe the methodology of our experiments. First, we describe the general design and investigation purposes of the two experiments we perform. Afterwards, we state the details of the experiments, that result from the general design of our experiments.

1) General Design of the Experiments: We perform two different experiments studying Swellfish Privacy for our main use case, which is the continuous publishing of private `Sum` aggregates of power consumption streams. The experiments have different investigation purposes as follows.

Experiment 1 The investigation purpose of the first experiment is an average-case case analysis. That is, we examine which data utility we can expect when applying Swellfish Privacy in, e.g., a city, where every individual submits a privacy specification, and we have to protect all of them.

Experiment 2 As mentioned, current state-of-the art mechanism do not exploit hiding intervals by design. Thus, in the second experiment, we examine how data utility progresses over time, if we extend one of the privacy specifications above with a realistic secret, that is dominant with respect to length, power consumption and privacy level, but has a short hiding interval with respect to the life cycle of the stream.

We aim at a realistic setting and a fair comparison. Therefore, in both experiments, we aim at using (1) real-world data streams with different mean loads, (2) a set of realistic privacy specifications of private households, (3) compare the Swellfish mechanism to state-of-the art competitor mechanisms from literature, including approximate variants, and the usage of (4) common post-processing techniques that improve their data utility. In the remainder of this section, we describe how we deal with these challenges.

As all mechanisms we use in our evaluation are based on adding noise, we repeat every experiment for every combination of data stream and mechanism 20 times to eliminate statistical bias. Further, we measure data utility, as usual, by the help of the (mean) relative error (MRE). The higher the MRE is, the lower is the data utility.

2) Real-World Data Streams: As the noise scale is independed from the actual value of the aggregate, differentially private mechanisms are known to give highly different errors for aggregates of different height. Therefore, the streams we use for evaluation should feature different mean aggregated loads, to support the generality of our results. A data set that features this, and we therefore use for our experiments, is the GEFCom 2012 data set [12] already used in [6]. It is a real-world data set consisting of summed-up hourly loads (kW) of 21 different US zones over a period of 4.5 years. The zones have different aggregated mean loads. For instance, Zones 4 and 8 are the two zones with the smallest mean load (0.5 and 3.77 mW), while Zone 18 has with 213.57 mW the highest one. As discussed, to determine a good value for the approximation parameter δ for the approximate variants of the mechanisms evaluated, we need to know the number of individuals per zone. However, this number is not given for the data set. Therefore, we use $\delta = \frac{1}{117{,}716{,}237}$ as suggested in [6][1]. To use the data streams for experiments, we have had to perform the following two prepossessing steps: First, originally, the data set was published in conjunction with a combined back- and fore-casting challenge. Therefore, the released data set is splitted in a train data set, and two solution data sets. We concatenated all three to obtain complete data streams over 4.5 years. Second, the data set contains hourly data, but our generated privacy specifications have 15 minutes resolution, as we discuss in the remainder. As the latter is more realistic, we re-sampled the GEFCom streams to 15 minutes resolution by linear interpolation.

3) Generation of Realistic Privacy Specifications: For our experiments, we need a set of realistic privacy specifications of private households. Realistic means that (1) secrets should correspond to events that are worth to hide in a given hiding interval and (2) the spectrum of private households should be

[1] Note that this value is actually too large, as $\frac{1}{|D|}$ it is not negligible in $|D|$. However, as we use the same approximation parameter for all mechanism, the choice of δ is not crucial for comparing them.

TABLE II
MECHANISMS COMPARED IN OUR EXPERIMENTS.

	Pure Privacy	Approx. Privacy
Swellfish	🟥 Swellfish(($\mathcal{P}_1, .., \mathcal{P}_n$)	▦ Swellfish($\mathcal{P}_1, .., \mathcal{P}_n$,$\delta$)
user-level DP	🟦Uniform($w = \mathcal{T}$,ϵ, $\Delta_{\mathbb{Q}} = 48.0$)	▦ Uniform($w := \mathcal{T}$,ϵ, δ, $\Delta_{\mathbb{Q}} = 48.0$)
w-event DP	🟩 Uniform(w,ϵ, $\Delta_{\mathbb{Q}}$) 🟧 RescueDP(w,ϵ,$\Delta_{\mathbb{Q}}$)	▦ Uniform(w,ϵ,δ, $\Delta_{\mathbb{Q}}$) -

TABLE III
PARAMETERS OF OUR COMPETITORS OBTAINED BY THEOREM 9.

	w	ϵ	$\Delta_{\mathbb{Q}}$	δ
Experiment 1	65	0.1	3.92	$\frac{1}{117{,}716{,}237}$
Experiment 2	672	0.1	3.92	$\frac{1}{117{,}716{,}237}$

realistic, i.e., they should have different characteristics (e.g., no. of inhabitants) and the distribution of the characteristics should match reality. In addition, we need a secret that is dominant with respect to this set, but only relevant for a short term compared to the lifetime of the streams for Experiment 2. However, literature reveals that no such set exists so far. So, we generate such a set that is realistic in the above sense, as well as the dominant secret for Experiment 2, as follows.

Realistic Set of Privacy Specifications: Reference [24] proposes a data generator, that generates a realistic set of households of different characteristics and their respective appliance usage. We implemented this data generator and generated 250 households with respective appliance usages over 4.5 years, which is the same time period that the GEFCom data set features. Then, for every household, we generate one privacy specification as follows. *Every* appliance usage cycle (except of fridge and refrigerator, which do not correspond to activities [24]) becomes one secret. The power and the length of the event corresponds to the values given by the generator. Further, for each secret, we sample the privacy level ϵ randomly from $[0.1, 1.0]$, and specify the hiding interval $\mathcal{J}$ by $\mathcal{J} = [\text{cycle\_start} - \tau \cdot T \cdot 2, \text{cycle\_end} + \tau \cdot T \cdot 2]$, where [cycle_start, cycle_end] is the time interval the appliance actually runs according to the data generator. By this, we generated a set 250 different privacy specifications. We argue that this set is realistic, as the generator is realistic. For reproducibility, the specifications are publicly available.[2].

Dominant Secret: We model the needed dominant secret by a one-week long event ($T = 24 \cdot 4 \cdot 7$), with $p = 3.92$, which is the maximum temporal sensitivity of the privacy specifications generated. Further, we specify $\epsilon = 0.1$ and $\mathcal{J}$ as the first week of the GEFCom data set. This event reflects holidays over one week, and the aim of the individual is to hide its occupancy. It is dominant compared to the privacy specifications generated.

4) Competitors: Table II states our competitors. As stated in Theorem 7, every w-event mechanism satisfies Swellfish Privacy, if the parameters are set like given in the theorem. Therefore, we use two w-event mechanisms, the Uniform and RescueDP mechanism already introduced in Section II-C2, as competitors. For completeness, we also compare ourselves against the Uniform mechanism that implements user-level differential privacy. We are in particular interested in the differences between pure and approximate mechanisms. Thus, for each competitor, we also included the approximate variant, if it exists. Note that for RescueDP, no approximate variant exists. We use Theorem 7 to determine the parameters of our competitors. See Table III for the resulting parameters. Observe that the parameters of both experiments differ in the window length w only. For the global sensitivity needed for user-level privacy, we use 48.0 kW, as suggested in [6]. It is the maximum possible consumption of a household in Germany.

5) Post-Processing: As stated in Theorem 4, Swellfish Privacy is immune to post-processing. If post-processing immunity is given, it is common to post-process the output of the mechanism to achieve higher data utility. Therefore, we perform two post-processing steps to the outputs of all mechanism except RescueDP, as this mechanism already involves post-processing steps. The two steps are *truncating*, to avoid negative aggregates, and *moving average filtering*, to smooth the stream. Selection criteria for the post-processing steps have been that (1) they can be applied in the streaming setting and (2) have already proven its worth for power consumption data, as it is not the main goal of this paper to elaborate on useful post-processing steps. The details of the steps are as follows:

Truncating All mechanisms considered in the evaluation rely on the addition of Laplace noise. However, this noise could be negative, which might result in negative aggregated values in the sanitized streams. Since power consumption is always zero or positive, we negative values round to zero.

Filtering We filter the data with a moving average filter. However, in preliminary experiments by ours revealed that (1) the filter improves data utility only if the noise scale is high enough, i.e., exceeds a certain threshold, and (2) that this threshold depends on the stream, i.e., GEFCom zone. Therefore, in a preliminary study, we determined reasonable noise scale thresholds for every zone, and apply this post-processing step for a combination of zone and mechanism only, if the expected noise scale exceeds this threshold. While the publication of such threshold violates Swellfish Privacy, the use of already published thresholds does not. As filter size, we use one day.

B. Results

In this section, we present and interpret the results of our two experiments. We start with Experiment 1 containing the average case analysis followed by Experiment 2 investigating on the effect of dominant secrets, that are only relevant for a relatively short time period. For either experiment, we validated all obtained results by comparing the relative errors with the retrieved noise scales. All results are in line with that. Further, all results are in line with the theoretic utility

[2] https://git.scc.kit.edu/nb0387/swellfish-public

analysis performed in Section IV-D increasing the validity of the experimental results.

1) Experiment 1: In the first experiment, we perform an average-case analysis, where we aim at protecting our generated realistic set of privacy specifications.

Result Presentation: Figure 3 shows the MRE of all mechanisms for every zone. Note that the y-axis of this and the following graphs has log scale. For illustration, we limited the graph by an MRE of 1,000%.

Ranking of the Mechanisms Due to the different mean loads, all mechanisms provide highly-different data utility for every zone. For instance, the error rates for zone 4 having the smallest mean load are the highest ones for each mechanism. However, the relative ranking of the mechanism is the same for every zone and is as follows. With small mean relative errors between 0.0064% (zone 18) and 2.95% (zone 4), the pure Swellfish mechanism has the lowest MRE, i.e., the highest data utility. The best competitor, the pure Uniform w-event mechanism, provides mean relative errors between 1.35% (zone 18) and 73.75% (zone 4). In contrast, the pure and approximate user-level mechanisms lead to the highest error by far, which is higher than 100% for every zone. RescueDP lies in the midfield with error rates between 4.5% (zone 3) and 69.3% (zone 4).

Pure vs. Approximate Except for user-level privacy, the errors of the pure and corresponding approximate mechanism are similar, while the pure one provides always slightly smaller error.

Result interpretation: The results indicate that in a realistic average-case setting, our Swellfish mechanism offers better data utility than any w-event mechanism by an order of an magnitude. This suggests that temporally varying privacy specifications improve data utility even in the average case. Further, preliminary results [5] suggest that RescueDP achieves higher data utility than the Uniform w-event mechanism for `Count` and `Histogram` aggregates. In contrast, our result indicate that this is not the case for `Sum` aggregates over power consumption streams. Further, as expected, the user-level mechanism do not provide any useful aggregate, as their error is always $> 100\%$. The indistinguishability of the error values between pure and respective approximate mechanism is expected, as the approximate mechanism only gain advantage from a large T, but here $T \leq 65$ holds.

2) Experiment 2: In the second experiment, we included the dominant secret. It is only relevant in the first week of the 4.5 years long streams, and aim at evaluating its effect.

Result Presentation: Figure 4 graphs the results of our second experiment. It shows the progress of the relative error over time for the pure and approximate mechanisms separately. Every line in each graph corresponds to one zone. The scaling is the same for the graphs in one row, but highly different for different rows. For space and illustrative reasons, we (1) only plot the first 12 weeks of the streams, as the trend is similar for the remaining ones, and (2) omit zone 4 due to very high error values. Overall, the results look valid, as the relative error rates of all mechanism follow the shape of the underlying aggregate streams. The only exception is RescueDP, that initially achieves small errors, which become quickly very large. We hypothesize that the adaptive budget allocation strategy of RescueDP is optimistic, resulting in small initial errors, quickly becoming large, if w is long.

Influence of the Dominant Secret The dominant secret leads to (1) higher noise scales of the Swellfish mechanism in the first week and (2) a higher window size w for the w-event mechanism. The parameters of the user-level mechanism do not change. As result, the relative error of the Swellfish mechanism is higher in the first week than in the remaining ones, and the relative error for the w-event mechanism is higher on average than in the first experiment. While it is higher by a constant factor for the Uniform mechanism, as the noise scale is proportional to w, the error of ResuceDP is by orders of magnitude worse than in the first experiment. Surprisingly, the relative error of the Swellfish mechanism is in the first week higher than the one of the Uniform w-event mechanism. However, we verified that in this week, the noise scale is similar to the one of Uniform w-event mechanism, and the difference is only due to different post-processing steps performed[3].

Pure vs. Approximate When comparing the pure and approximate mechanism, it turns out that for every mechanism, the approximate variant has better data utility by orders of magnitude. For Swellfish mechanism, this holds for the first week. For the other mechanism, during the whole lifetime of the stream, while remaining above .

Result Interpretation: The results show that the Swellfish mechanism is the only mechanism where the dominant event has only a local influence, meaning that the *required* error is only observable in times, where the secret is relevant, i.e., the higher error is actually required. In contrast, the dominant secret worsens data utility of the w-event mechanism by orders of magnitude during the whole lifetime of the stream.

C. Summary

Based on the results of both experiments, we have empirically shown that exploiting temporally varying privacy specifications results in high data utility. Specifically, the utility rates that are an order of magnitude better than the ones of all competitors, including the approximate ones. This holds for the average case, and is even more visible in case we have to hide a dominant secret only relevant for a short time. In the latter case, our approximate Swellfish mechanism improves the data utility of our pure mechanism further. We performed our experiments on a set of 20 real-world power consumption streams having different mean loads, and used realistic privacy specifications, which indicates the generality of our results.

[3] In contrast to the output of the Uniform w-event mechanism we did not apply the moving average filter to the output of the Swellfish mechanism, as the mean noise scale provided Swellfish mechanism we used as criteria to decide if we apply the filter, is too small.

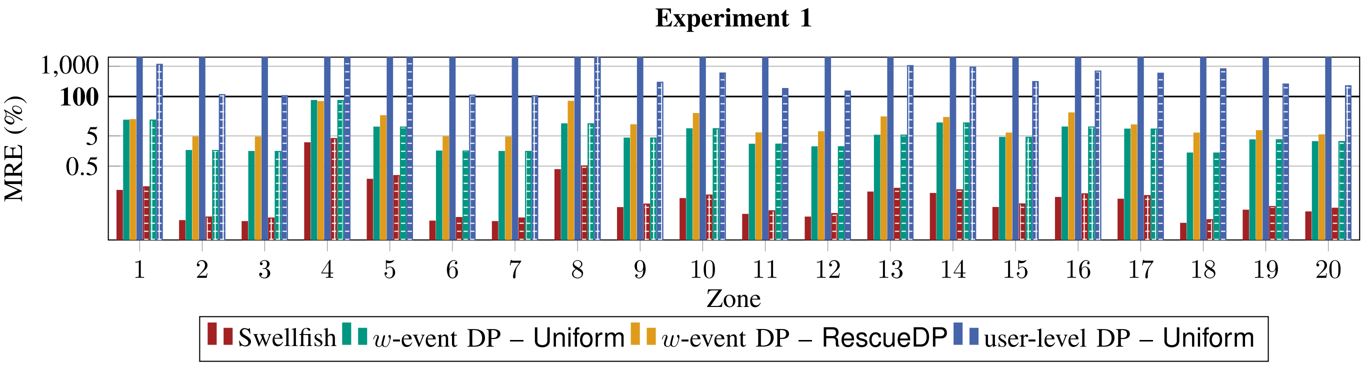

Fig. 3. Results of Experiment 1 for each zone.

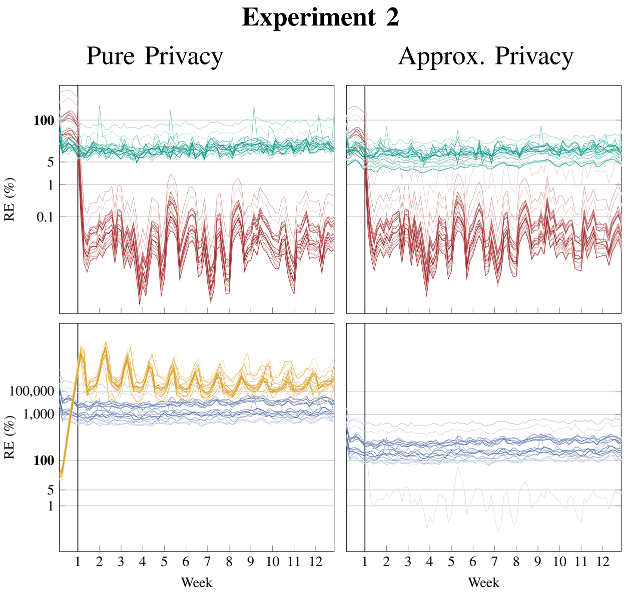

Fig. 4. Results of Experiment 2.

VII. Conclusions and Future Work

We propose Swellfish Privacy, a privacy framework for the continuous publishing of differentially private `Sum` aggregates, together with a respective mechanism. It is a generalization of the w-event differential privacy framework offering better data utility by exploiting our concept of temporally varying privacy specifications. With a theoretic utility analysis, as well as two experiments on real-world power consumption streams with realistic privacy specifications, we reveal that the Swellfish mechanism outperforms known competitors by at least an order of magnitude. This holds in the average case, and becomes even more visible in the presence of dominant secrets.

We see various directions for future work. First, generally, privacy specifications are considered as public information. However, this might be critical, as privacy specification ifself might contain private information. To deal with that, we plan to investigate whether we could rely on distributed noise generation, encryption or local differential privacy. Second, the results of our evaluation suggest to investigate systematically how to change the adaptive budget allocation strategy of RescueDP such that it is able to deal with high values of w. Third, the power consumption values of an individual are generally temporally correlated. In the future, we aim at exploiting these correlations to improve the data utility of our mechanism further, e.g., by adaptive sampling or budget allocation.

References

[1] S. Muthukrishnan *et al.*, “Data streams: Algorithms and applications,” *Foundations and Trends® in Theoretical Computer Science*, 2005.

[2] L. Fan and L. Xiong, “An adaptive approach to real-time aggregate monitoring with differential privacy,” *IEEE TKDE*, 2014.

[3] G. Kellaris *et al.*, “Differentially private event sequences over infinite streams,” *PVLDB*, 2014.

[4] H. Li *et al.*, “Differentially private histogram publication for dynamic datasets: an adaptive sampling approach,” in *CIKM*. ACM, 2015.

[5] Q. Wang *et al.*, “Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy,” *TDSC*, 2018.

[6] G. Eibl *et al.*, “The influence of differential privacy on short term electric load forecasting,” *Energy Informatics*, 2018.

[7] C. Dwork *et al.*, “The algorithmic foundations of differential privacy,” *Found. Trends Theor. Comput. Sci.*, 2014.

[8] C. Dwork, “Differential Privacy,” in *Encyclopedia of Cryptography and Security*. Springer, 2011.

[9] D. Kifer and A. Machanavajjhala, “Pufferfish: A Framework for Mathematical Privacy Definitions,” *TODS*, 2014.

[10] X. He, A. Machanavajjhala, and B. Ding, “Blowfish Privacy: Tuning Privacy-Utility Trade-Offs Using Policies,” in *SIGMOD*. ACM, 2014.

[11] Z. Jorgensen, T. Yu, and G. Cormode, “Conservative or liberal? personalized differential privacy,” in *ICDE*. IEEE, 2015.

[12] T. Hong, P. Pinson, and S. Fan, “Global energy forecasting competition 2012,” *Int. J. Forecast.*, 2014.

[13] G. Ács and C. Castelluccia, “I have a dream! (differentially private smart metering),” in *IH*. Springer, 2011.

[14] P. Kairouz, S. Oh, and P. Viswanath, “The composition theorem for differential privacy,” *IEEE Trans. Inf. Theory*, 2017.

[15] K. Chatzikokolakis *et al.*, “Broadening the scope of differential privacy using metrics,” in *PETS*, 2013.

[16] F. D. McSherry, “Privacy integrated queries: an extensible platform for privacy-preserving data analysis,” in *SIGMOD*. ACM, 2009.

[17] L. Sweeney, “k-anonymity: A model for protecting privacy,” *Int’l. J. of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2002.

[18] D. Kifer and B.-R. Lin, “An axiomatic view of statistical privacy and utility,” *Journal of Privacy and Confidentiality*, 2012.

[19] W. Kleiminger *et al.*, “Occupancy detection from electricity consumption data,” in *BuildSys*. ACM, 2013.

[20] G. Hart, “Nonintrusive appliance load monitoring,” *Proc. IEEE*, 1992.

[21] F. Laforet, E. Buchmann, and K. Böhm, “Individual privacy constraints on time-series data,” *Information Systems*, 2015.

[22] C. Tex *et al.*, "PrivEnergy – a privacy operator framework addressing individual concerns," in *e-Energy*. ACM, 2018.

[23] C. Dwork *et al.*, "Our data, ourselves: Privacy via distributed noise generation," in *EUROCRYPT*. Springer, 2006.

[24] S. Gottwalt *et al.*, "Demand side managementa simulation of household behavior under variable prices," *Energy Policy*, 2011.