# P2KMV: A Privacy-preserving Counting Sketch for Efficient and Accurate Set Intersection Cardinality Estimations

HAGEN SPARKA, Humboldt University of Berlin
FLORIAN TSCHORSCH, Technical University of Berlin
BJÖRN SCHEUERMANN, Humboldt University of Berlin

In this paper, we propose P2KMV, a novel privacy-preserving counting sketch, based on the $k$ minimum values algorithm. With P2KMV, we offer a versatile privacy-enhanced technology for obtaining statistics, following the principle of data minimization, and aiming for the sweet spot between privacy, accuracy, and computational efficiency. As our main contribution, we develop methods to perform set operations, which facilitate cardinality estimates under strong privacy requirements. Most notably, we propose an efficient, privacy-preserving algorithm to estimate the set intersection cardinality. P2KMV provides plausible deniability for all data items contained in the sketch. We discuss the algorithm's privacy guarantees as well as the accuracy of the obtained estimates. An experimental evaluation confirms our analytical expectations and provides insights regarding parameter choices.

## 1 INTRODUCTION

In this paper, we investigate how to obtain privacy-preserving user statistics. We consider the scenario of a central service or application which—in one way or the other—obtains personal identifiable information from its users. This service aims to calculate statistics from these data, across users. Examples might be how many distinct users accessed the service using software version X, or how many distinct users issued requests of type Y, or how many distinct users satisfy both these criteria.

However, the service that we consider also aims to *not* store all the personal details for answering such questions to protect its users from potential data leaks or breaches. The storage of data, which can be associated with individual users, poses a privacy risk in itself. Anyone with access to the data, be it an attacker gaining illegitimate access to the system or a government agency requesting data, can (mis-)use this information.

A centralized system as outlined above reflects today's (and most likely tomorrow's) common practice. However, it unfortunately makes approaches such as multi-party computation based on secret sharing [30, 34, 39] or homomorphic encryption [26, 28] inapplicable. Likewise, advanced differential privacy approaches [10, 33] do not provide a solution to this scenario. In case of a subpoena, for example, encrypted data sets can be recovered. Even in a distributed setting with multiple independent entities the data remains recoverable, if the entities cooperate.

The key question considered here is therefore how statistics can be collected in a central place—without storing detailed information on individual users or operations. Most notably, we consider estimating the size of set intersections in a privacy-preserving manner. This can be used, for instance, to determine correlations and overlaps between groups of users.

Probabilistic counting sketches have recently been identified to serve as a privacy-enhancing technology (PET) for obtaining statistics [30, 32, 40]. While calculating unions using counting sketches is often straightforward, we argue that none of the existing solutions is well suited for estimating set intersection cardinalities. Estimating intersection cardinalities is a challenge because

it typically requires combining many individual estimates, so that the results tend to have poor accuracy due to error propagation.

In this paper, we contribute a novel algorithm, P2KMV, and demonstrate that accurate statistics and good privacy are—even in a centralized setting—not mutually exclusive. With P2KMV, we take the idea of *data minimization*, i. e., collecting as little personal data as possible, a step forward: we show how to obtain certain aggregate statistics, while storing a small data sample only. In order to protect *all* users, including identifiable information in the data samples, we additionally generate *provable* plausible deniability.

To this end, we build upon the so-called $k$ minimum values (KMV) counting sketch [3] and design a privacy extension that retains the full feature set of the original KMV sketch. While we use a data perturbation technique, which adds random noise to KMV sketches to protect privacy, our approach is able to produce accurate set intersection cardinality estimates, even when intersecting many sets. Therefore, P2KMV complements the set of PETs and provides a missing feature for practical deployment.

To show P2KMV's merits, we evaluate it in a controlled deterministic simulation environment and compare P2KMV to related approaches—namely, probabilistic counting with stochastic averaging (PCSA) as in [20], privacy-enhanced PCSA as in [40], and a Bloom filter-based approach as introduced in [30]. Our evaluations show that significantly higher accuracy is achievable especially for larger numbers of intersected sets. P2KMV inherits the high accuracy and efficiency of KMV when estimating set intersection cardinalities, and at the same time guarantees enhanced privacy, even against an adversary with pre-knowledge.

The contributions of our paper are summarized as follows:

- We consider a realistic and strong threat model in Section 2, which goes beyond a typical honest-but-curious adversary and includes state-level and external adversaries.
- We identify KMV sketches as a basis for efficient cardinality estimates and set operations, particularly for set intersection cardinality estimation. As we argue in Section 3, the latter is necessary for elaborate statistics but can become computationally expensive.
- We propose P2KMV in Section 4. We formally derive how P2KMV can provide enhanced privacy as well as accurate estimates. We also evaluate the influence of an adversary's pre-knowledge on P2KMV's privacy and prove that P2KMV provides plausible deniability for every user.
- We analyze P2KMV's accuracy in Section 5. We compare it to related approaches, and show that P2KMV's set intersection cardinality estimation is computationally efficient. As a result from our in-depth evaluation, we derive guidelines to chose suitable parameter values for P2KMV.

These ideas offer a versatile PET that can be applied to many use cases. Section 6 discusses related work and emphasizes the novelty of the approach. In summary, P2KMV balances the trade-off between accuracy, efficiency, and privacy.

## 2 SYSTEM AND THREAT MODEL

In this section, we outline our system and threat model. In particular, we define the involved entities and classify the adversary's capabilities. The specific implementation details are subject of the following sections.

Our system model consists of a set of users and a service provider. Figure 1 illustrates these entities and their relation in our system. On a high level, users contact a service or use an application and generate data to be collected. We assume that every user has a personal ID. The service provider learns the respective user ID with each user access. For example, a user could use a messaging

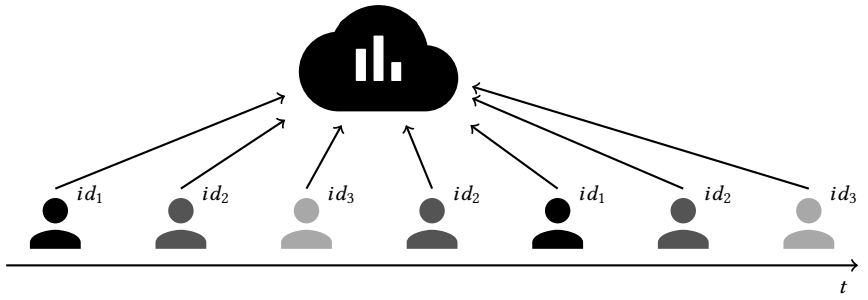



Fig. 1. In our system model, recurring users access a centralized service; the service provider collects and analyzes user data.

service to communicate with other users. In this case, the ID could be the user's cell phone number or email address. It could also be an IP address or any other unique identifier. The messaging service provider, who forwards messages, learns the ID and can link data items to this ID.

The service provider is a central data collector and data analyst at the same time. Statistics, which might be of interest for the service provider, include but are not limited to the usage share of operating systems or the adoption of a software update. For example, the service provider might want to answer the question "how many distinct users use software version X on system Y?" As consequence, any statistical framework should provide means of data associations.

In order to derive meaningful statistics, the service provider needs to collect some data. A common approach is to store all available information. However, storing personal identifiable information poses a privacy risk. While encrypting the data somewhat improves the situation, it cannot eliminate the privacy risk completely: the data can still be recovered. An external adversary gaining access to the system and the cryptographic key material can decrypt the data. Likewise, a state-level adversary can force the service provider to reveal the data. To our best knowledge, there is no cryptographic primitive that mitigates such a threat.

Signal, a secure instant messaging service, experienced a comparable situation [35]. The developers received a subpoena requesting to provide information about two Signal users. Signal, however, minimized the retained data about its users; the only information they could provide was the date and time of registration and the last date of a user's connectivity to the Signal service. From a privacy perspective storing virtually no data at all provides the strongest privacy. Nevertheless, there are good reasons for obtaining user statistics. One might resort to producing highly aggregated statistics only, without actually storing any raw data. This approach comes with heavy limitations, though. For instance, aggregated statistics would lead to gross overestimation as it is not possible to count *distinct users* only.

In our system model, we also refrain from assuming user interaction in the process of data collection and opt for a more general approach. While user interaction, as in the case of randomized response schemes [17, 43], leads to strong privacy guarantees, it is sometimes not practical nor feasible. Metadata, e.g., a software version, are relevant sources for statistics but are often an inherent part of a communication protocol. Thus, they can be considered immutable by the user, which makes it infeasible to assume that a user can add "noise" to the data collection by altering values. Instead, we assume a passive/implicit data collection.

In order to allow more sophisticated statistics without the need to store fine-grained personal data, we aim for a data minimization technique that reduces the amount of stored data and still

provides meaningful statistics. To this end, we use a probabilistic data structure which discards the majority of data items and stores a small data sample only. Despite this fact, the estimates remain accurate. This significantly reduces the amount of personal identifiable information. The data samples can, however, still be linked to users. We assume that a user's absence, e. g., not using a service, is of no concern and hence concentrate on preventing an adversary from being able to infer whether a certain user is part of a statistic. We therefore extend our approach and propose a solution that uses data perturbation to guarantee *plausible deniability* for every user.

Our threat model assumes an adversary that does not manipulate the data, but is able to gain access to the data—either by entering the system or by issuing an order. The threat model, therefore, includes external, internal, and state-level adversaries alike. We explicitly exclude the service provider from our threat model, though, because the service provider could tap the data anyway. Therefore, we have to trust the service provider. We further assume that the adversary is able to retrieve snapshots of the data only and is not able to monitor the service over a longer period. Otherwise, the adversary would take a similar role as the service provider, i. e., being able to tap the raw data. Our adversary, therefore, shares similar characteristics with a covert attacker, who usually aims for transferring/extracting data but at the same time wants to remain hidden.

With our approach, personal identifiable information is stored in a way that it cannot be recovered. All sensitive material, e. g., the perturbation, which helps to protect privacy, are designed to be completely volatile. In summary, we envisage a privacy-enhancing technology for user statistics that thwarts linking users on the basis of stored data in the presence of external and even state-level adversaries.

## 3 USER STATISTICS WITH KMV

In this section, we argue that KMV [3]—the basis for our own algorithm P2KMV—can be used to combine data minimization with accurate and efficient set operations. Here, we first recapitulate a number of properties which directly emerge from KMV's design. In particular, we generalize the Jaccard similarity coefficient for estimating set intersection cardinalities with KMV. This will subsequently lead to the specific algorithmic tools and choices for P2KMV, and therefore to our main contributions.

### 3.1 Data Minimization with KMV

KMV was originally devised to efficiently estimate the cardinality of elements in a data stream. The algorithm belongs to the family of probabilistic counting sketches. For the cardinality estimation of a data set a succinct representation—the KMV sketch—is constructed by hashing each element and storing the $k$ smallest distinct hash values only. We also say that an element has been *sampled* by the sketch. In the following, we will denote the set of the $k$ smallest hash values as $K$.

Retaining only the $k$ smallest hash values already reduces the amount of personal identifiable information that is stored and thus increases privacy—though, as we will see, this is still far from perfect.

The hash values can be either chosen uniformly from $(0, 1)$ as in [3] or from the integers in $[1, N]$ as in [6]. In the remainder of this paper, we use the latter approach and define $N$ as $n_{\mathrm{ID}}$, the size of set $\mathcal{I}$, which contains all IDs in the system. The hash values are calculated using a hash function $H$, with the following properties (a) $H : \mathcal{I} \to [1, n_{\mathrm{ID}}]$ uniformly maps each ID in the system to an integer in $[1, n_{\mathrm{ID}}]$, (b) all hash values are chosen independently, and (c) $H$ is injective. In practice, such a hash function, which actually is a special type of permutation, can be constructed using a cryptographic hash function, such as SHA-256: According to [6], when using a hash function $f$ whose codomain is much larger than its domain, e. g., a codomain whose size is the square of

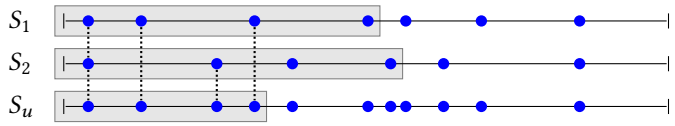



Fig. 2. KMV sketch and union visualization ($k = 4$).

the domain's size, this function already satisfies (b) and (c) with high probability. We then can use $f$ to generate a hash function which also satisfies (a) by hashing all IDs and using the order of their hash values as follows: let $(h_1, h_2, \ldots, h_{n_{\mathrm{ID}}})$ be the sorted tuple of hash values generated by applying $f$ to all IDs in $\mathcal{I}$. Then we define $H$ as the function mapping each ID $x$ to the index $i$ of the hash value $h_i$ in $(h_1, h_2, \ldots, h_{n_{\mathrm{ID}}})$, where $h_i = f(x)$.

### 3.2 Statistics with Counting Sketches

Counting sketches can be used to collect statistics on categorical variables by having each sketch represent a certain category. For instance, categorical values could be a software version or an operating system. A service provider asking "how many distinct users run software version X" needs to know the cardinality of X's user set. To learn this, they may hash each user's ID into a KMV sketch maintained for this software version. Note that, if the same users with software version X accesses the service again, the user will not be counted twice, because adding the respective ID twice does not change the KMV data structure.

This duplicate insensitivity is an inherent feature of counting sketches that makes them also suitable to collect population and distribution statistics. Collecting the software version distribution, for example, would require an individual sketch for each possible software version. Basically, everything that can be enumerated can be counted with sketches. Since sketches follow the idea of data minimization, their storage demand is reasonable, also for very large number of sets.

As we show in the following, counting sketches are representatives of a set and therefore obey the algebra of sets. In particular, they satisfy the fundamental binary operations of set union and intersection. For instance, this can be used to aggregate individual observations of a variable into groups.

### 3.3 Set Cardinality Estimation

The number of unique elements in a set $M$ can be estimated by taking $K$, the $k$ smallest hash values, into consideration. If there are less than $k$ unique elements in $M$, the cardinality is exactly $|K|$. For sets with more unique elements, let $\max(K)$ be the largest hash value in $K$. Note that $\max(K)$ will typically decrease with an increasing number of distinct elements added to the sketch, because only the smallest hash values are maintained. Since $H$ maps the IDs uniformly to its codomain, the interval $[1, \max(K)]$ can be used as a representative for the whole codomain. The set cardinality can therefore be estimated as $\hat{\vartheta}_{|M|}$ by

$$\hat{\vartheta}_{|M|} = \frac{k}{\max(K)} \cdot n_{\mathrm{ID}}. \tag{1}$$

So, by evaluating the sketch, an estimate for the number of distinct users can be obtained. In order to count many statistics at once, the service provider holds multiple sketches in parallel, each representing a category of interest.

### 3.4 Set Union Cardinality Estimation

Along similar lines, the cardinality of a union of sets can be estimated. Let $k_1, k_2, \ldots, k_n$ be the maximum numbers of hash values stored by the KMV sketches $S_1, S_2, \ldots, S_n$ that use the same hash function $H$. In order to estimate the union, we construct a KMV sketch $S_u$ with $k_u = \min(k_1, k_2, \ldots, k_n)$. The new sketch stores the $k_u$ smallest hash values from $K_1, K_2, \ldots, K_n$. As a consequence, $S_u$ contains the very same smallest $k_u$ values that would have been stored if all IDs were sampled directly by $S_u$. We denote this set of $S_u$'s $k_u$ smallest hash values as $K_u$. Note that the construction of $S_u$ ensures that for each hash value $h$ in $K_u$, $h \leq max(K_i)$ where $i \in \{1, 2, \ldots, n\}$, holds. Figure 2 illustrates this process for two KMV sketches $S_1$ and $S_2$. Evaluating $S_u$ yields a set union cardinality estimate.

### 3.5 Set Intersection Cardinality

Set intersection cardinalities can be used to calculate associations between items, e. g. [22]. For instance, a service provider may be interested in learning associations such as "how many distinct users use software version X on system Y?". If we know that we are interested in this kind of statistic in advance *and* are able to observe X and Y at the same time, we could instantiate a dedicated sketch, which we use to count users matching both characteristics. Even if we are able to determine all kinds of relevant statistics in advance, though, X and Y might be in separate requests or messages, which prevents us from associating these characteristics on the fly. We, therefore, argue that this approach is neither flexible nor practical. Thus, intersections complement the necessary set of primitives to produce meaningful statistics.

*3.5.1 Principle of Inclusion-Exclusion.* Most counting sketches inherently allow to estimate cardinalities of sets and their unions only. However, the size of the set intersection can—in general—be estimated by using the inclusion-exclusion principle. In its simplest instance the principle states that for two sets $M_1$ and $M_2$ the size of their intersection is $|M_1 \cap M_2| = |M_1| + |M_2| - |M_1 \cup M_2|$. It can be generalized to higher numbers of sets. For $n$ sets $M_1, M_2, \ldots, M_n$, the general inclusion-exclusion principle [18] can be used to define the set intersection cardinality recursively as

$$\left|\bigcap_{i=1}^{n} M_i\right| = (-1)^{n+1}\left|\bigcup_{j=1}^{n} M_j\right| - \sum_{k=1}^{n-1}(-1)^{n+k} \sum_{\substack{Z \subseteq \{1,2,\ldots,n\} \\ |Z|=k}} \left|\bigcap_{l \in Z} M_l\right|.$$

For $n = 2$, the formula stated before follows.

For increasing $n$, however, the inclusion-exclusion principle quickly becomes prohibitively expensive in terms of computation effort. In general, for $n$ sets all intersection cardinalities of the $n$ subsets with size $n - 1$ are necessary. Since these $n$ intersection cardinalities are not known in advance, they all have to be calculated accordingly. Therefore, virtually all set intersection cardinalities in the power set of $\{M_1, M_2, \ldots, M_n\}$ need to be calculated, resulting in computation times that grow exponentially with $n$. Moreover, errors of individual estimates tend to sum up, resulting in low accuracy.

*3.5.2 Jaccard Similarity Coefficient.* One big strength of KMV, compared to other counting sketches, is its ability to estimate set intersection cardinalities efficiently and accurately without the need to rely on the inclusion-exclusion principle, thereby circumventing the problem described above. Beyer et al. [6] used an alternative approach to the inclusion-exclusion principle to estimate the intersection cardinality of two KMV sketches. They observed: if two KMV sketches $S_1$ and $S_2$ use the same hash function $H$, the ratio between the intersection size and the union size of their $k$

smallest hash values is (approximately) the same as the respective ratio in the underlying sets. This ratio is called the *Jaccard index* or *Jaccard similarity coefficient*.

Assume for example two sets $|M_1| = 800$ and $|M_2| = 1300$ with $|M_1 \cap M_2| = 100$ and consequently $|M_1 \cup M_2| = 2000$. Let $S_1, S_2$, and $S_u$ denote KMV sketches for $M_1, M_2$, and $M_1 \cup M_2$, respectively. Further let $k_u$ be 400. $K_u$, the set of the $k_u$ smallest hash values in $S_u$, will then hold about 20 hash values that correspond to IDs in $M_1 \cap M_2$. This subset of $K_u$ is called $C_0$. Given $S_1, S_2$ and $S_u$ one can directly derive $C_0$. Then, the intersection cardinality can be estimated by multiplying $|C_0|/k_u$ with $|M_1 \cup M_2|$, i. e., $20/400 \cdot 2000 = 100$. The fraction $|C_0|/k_u$ is the (estimated) Jaccard index.

Here, we generalize Beyer et al.'s approach for $n > 2$ sets. Just like for two sets, the Jaccard index $J$ for the sets $M_1, M_2, \ldots, M_n$ describes the ratio between the sets' intersection and union cardinality. It is defined as

$$J(M_1, M_2, \ldots, M_n) = \left| \bigcap_{i=1}^{n} M_i \right| \Big/ \left| \bigcup_{i=1}^{n} M_i \right|.$$

In order to estimate the cardinality of the intersection the following steps must be taken: first, calculate $S_u$ from $S_1, S_2, \ldots, S_n$ as described above. Second, determine $|C_0|$ by counting the number of distinct hash values that are stored in $S_1, S_2, \ldots, S_n$, *and* $S_u$. Third, the Jaccard index $J(M_1, M_2, \ldots, M_n)$ can be estimated by

$$\hat{\vartheta}_J = \frac{|C_0|}{k_u}.$$

Finally, given $\hat{\vartheta}_J$ and the set union cardinality estimate $\hat{\vartheta}_{|U|}$ (obtained by evaluating $S_u$), the set intersection cardinality can be estimated by

$$\hat{\vartheta}_{|I|} = \hat{\vartheta}_J \cdot \hat{\vartheta}_{|U|}. \tag{2}$$

Note that, unlike with the inclusion-exclusion principle, there is no exponential cost explosion associated with increasing the number of sets that are intersected.

This—so far admittedly rather simple—generalization paves the ground for turning towards further measures for privacy protection, and therefore for our main contributions, in the next section. As will become clear, set intersection estimates based on the Jaccard index will remain possible and accurate, but the details will become significantly more intricate.

## 4 PRIVACY-PRESERVING KMV

As shown so far, KMV is a well-suited basis for estimating the cardinality of set intersections. Unmodified KMV significantly reduces the amount of stored personal identifiable information by keeping only the $k$ smallest hash values. These, however, can still be linked to specific IDs by an adversary. Consequently, privacy is preserved for most users—but compromised for the subset of users with the smallest hash values.

To improve on this, we first formally discuss how much information an adversary can gain about individual IDs by investigating a KMV sketch. As outlined in our threat model, we assume that an adversary gained access to the counting sketch and can investigate the $k$ smallest values. We further assume that an adversary knows the algorithm specifications, particularly the hash function $H$, the hash salt, and the set of candidate IDs (so that their hash values can be enumerated). We want to prevent an adversary from concluding that a certain ID is part of a data set. Our model also captures that adversaries bring previous knowledge; in particular, they can already quantify a probability that a certain ID is part of the set.

The challenge here is, while protecting privacy, to still be able to produce accurate estimates. In the end, we aim for a solution that balances the trade-off between computational efficiency, accuracy, *and* privacy.

### 4.1 Privacy Extension

As stated above, unmodified KMV leaks private data: an adversary might be able to link the $k$ smallest hash values to IDs. To solve this problem, we introduce a perturbation technique, leading to a new privacy-preserving counting sketch, namely *Privacy-Preserving KMV* (P2KMV). It is related in spirit to what has been proposed for PCSA in [40]; however, the technique proposed here is not only based on a different type of sketch, but it also allows for Jaccard-index-based intersection estimates.

P2KMV obfuscates the $k$ smallest hash values by adding dummy hash values. During the initialization phase, each hash value is chosen independently with probability $p \in [0, 1]$ as a dummy element and added to the sketch. The $k$ smallest dummy hash values are used as the sketch's initial $k$ smallest hash values. By doing so, P2KMV also protects the IDs that are mapped to the $k$ smallest values: each of these hash values could be either a dummy element or originate from a sampled ID.

We can improve on this process by taken the following property into account: since each hash value will be chosen as a dummy independently with probability $p$, the distance between these values follows a geometric distribution with an expected value of $p^{-1}$. It will therefore suffice to randomly draw $k$ distances $d_1, d_2, \ldots, d_k$ from a geometric distribution and choose the dummies accordingly.

Since the probability $p$ determines the degree of obfuscation, we call it the *privacy level*. This privacy level allows us to guarantee *plausible deniability* for each ID sampled by a P2KMV sketch. There are different conflicting definitions of plausible deniability in literature, e. g. [2, 7]. Here, we say that a data structure provides plausible deniability, if some uncertainty about the fact that a specific ID $x$ has the property analyzed by this data structure, i. e. $x \in M$, will remain after inspection of the data structure.

We use the concept of pre- and post-knowledge, to further formalize this notion of plausible deniability: let $M$ be the set of IDs that were sampled by the P2KMV sketch, and let $x \in M$ be the adversary's ID of interest. We assume that the adversary has some pre-knowledge on how likely $x \in M$ is and that he wants to confirm that $x \in M$ holds. We denote the adversary's pre-knowledge as the a-priori probability $\Pr_{\text{pre}}(x)$. Accordingly, we denote the adversary's post-knowledge as the a-posteriori probability $\Pr_{\text{post}}(x)$, i. e., the adversary's knowledge on how likely $x \in M$ is *after* analyzing the sketch. With these notions we define plausible deniability as follows.

Definition 4.1 (Plausible Deniability). We state that a data structure provides ($\gamma$)-plausible deniability, if there is a $\gamma > 0$ such that for *any* set $M$ of IDs sampled by the data structure

$$\frac{1 - \Pr_{\text{post}}(x)}{1 - \Pr_{\text{pre}}(x)} \geq \gamma,$$

holds for all $x \in M$ and all $\Pr_{\text{pre}}(x) < 1$.

This definition is centered around the question, how well the data structure keeps an adversary from concluding that an ID $x$ is in $M$ as a function of the adversary's pre-knowledge. By using the lower bound of the fraction between the remaining uncertainty after and before analyzing the data structure for *all* IDs sampled by the data structure and *all* possible levels of pre-knowledge, $\gamma$ constitutes a very robust privacy metric. Note, that if the attacker can obtain complete certainty ($\Pr_{\text{post}}(x) = 1$) for any combination of pre-knowledge and ID $x \in M$, no $\gamma > 0$ can be found and there will be no plausible deniability by our definition.

Theorem 4.1 (plausible deniability of P2KMV). *A P2KMV sketch provides ($\gamma$)-plausible deniability, where $\gamma$ is its privacy level $p$.*

Proof. First, note that the adversary has a knowledge gain only if $H(x) \leq max(K)$, i. e., the hash value of $x$ is one of the $k$ smallest hash values. If $H(x) > max(K)$, then the sketches with and without $x$ being sampled do not differ, so that no information can be gained, resulting in the adversary learning *nothing* about $x$. In this case $\Pr_{\text{post}}(x) = \Pr_{\text{pre}}(x)$ holds.

Due to the dummy values, however, the adversary cannot be sure if $H(x)$ is indicative of $x$'s presence in $M$, even if $H(x) \in K$ holds. In this case, $\Pr_{\text{post}}(x)$ is equal to the probability that $x \in M$ given $H(x) \in K$. That is, $\Pr_{\text{post}}(x) = \Pr(x \in M | H(x) \in K)$.

Using Bayes' theorem, we can re-write $\Pr_{\text{post}}(x)$ as

$$\Pr_{\text{post}}(x) = \frac{\Pr(H(x) \in K | x \in M) \cdot \Pr(x \in M)}{\Pr(H(x) \in K)}. \tag{3}$$

As stated above, $x$ is only vulnerable when $H(x)$ is among the $k$ smallest values; then, $\Pr(H(x) \in K | x \in M)$ becomes one. Moreover, $\Pr(x \in M)$ is the adversary's pre-knowledge, i. e., equal to $\Pr_{\text{pre}}(x)$. Applying the law of total probability yields

$$\begin{aligned}\Pr(x \in K) = &\Pr(H(x) \in K | x \in M) \cdot \Pr(x \in M) \\ &+ \Pr(H(x) \in K | x \notin M) \cdot \Pr(x \notin M).\end{aligned}$$

Note that the first product is identical to the enumerator in Equation (3). The second product consists of $\Pr(x \notin M)$, which simply is $1 - \Pr_{\text{pre}}(x)$, and $\Pr(H(x) \in K | x \notin M)$. The latter describes the probability that $x$'s hash value is one of the $k$ smallest values, given that $x$ is not in $M$. This can only happen if $x$ is chosen as a dummy element, i. e., with probability $p$. Combining all insights yields

$$\Pr_{\text{post}}(x) = \frac{\Pr_{\text{pre}}(x)}{p + (1-p) \cdot \Pr_{\text{pre}}(x)}$$

as the *worst case* post-knowledge of an adversary analyzing a P2KMV sketch. Thus

$$\frac{1 - \Pr_{\text{post}}(x)}{1 - \Pr_{\text{pre}}(x)} \geq \frac{p}{p + (1-p) \cdot \Pr_{\text{pre}}(x)}$$

holds for every $x \in M$. The right part of this inequality decreases monotonically when the adversary's pre-knowledge increases. Since

$$\lim_{\Pr_{\text{pre}}(x) \to 1} \frac{p}{p + (1-p) \cdot \Pr_{\text{pre}}(x)} = p$$

for every $x \in M$, we can conclude that

$$\frac{1 - \Pr_{\text{post}}(x)}{1 - \Pr_{\text{pre}}(x)} \geq p.$$

Therefore a P2KMV sketch provides ($\gamma$)-plausible deniability with $\gamma = p$. □

Note that plausible deniability does not protect against an adversary whose goal is to find out, if $x \notin M$, i. e., an ID does *not* have the monitored property. We conclude that P2KMV is resistant against an adversary with access to the sketch by combining data minimization with perturbation. This way, P2KMV provides provable plausible deniability for *all* personal identifiable information.

### 4.2 Set Cardinality Estimation

While P2KMV effectively protects privacy, it is no longer possible to use KMV's estimations as presented earlier: the dummy elements would lead to gross overestimation. In the following, we derive new methodologies for estimating the set cardinality, set union cardinality, and set intersection cardinality using P2KMV. These estimation algorithms belong to the main contributions of this paper.

During the initialization of P2KMV, on average $p \cdot n_{\mathrm{ID}}$ hash values are chosen as dummy elements. However, one cannot tell which of the $k$ smallest hash values are dummy elements. This is at the heart of the algorithm's privacy protection. To correctly estimate the set cardinality, though, we have to remove the dummies' influence on the estimation.

Let $S$ be a P2KMV sketch with privacy level $p$. Further, let $D$ be the inserted dummy hash values, $H(M)$ the set of hash values of the IDs sampled by $S$ and $K$ the set of the $k$ smallest hash values. Because $H$ is injective, $|M| = |H(M)|$ holds. According to Equation (1), we can use $S$ to estimate $|H(M) \cup D|$. A naive way to estimate $H(M)$'s cardinality is to subtract $|D|$ from the estimate $\hat{\vartheta}_{|H(M)\cup D|}$. However, this yields an unbiased estimate only if $H(M) \cap D = \emptyset$. Since there might be "collisions", i. e., hash values of sampled IDs which had also been inserted as dummy elements, we have to assume that $H(M) \cap D$ is, in general, not empty.

The problem can be modeled as an urn problem with $|D|$ black and $n_{\mathrm{ID}} - |D|$ white marbles, which are drawn without replacement. The black marbles represent the initially inserted dummy elements. $|H(M)|$ marbles are drawn from the urn. The number of drawn white marbles $X$ follows a hypergeometric distribution; the expected number of white marbles is given by

$$E[X] = |H(M)| \cdot \frac{n_{\mathrm{ID}} - |D|}{n_{\mathrm{ID}}} \approx |H(M)| \cdot (1 - p),$$

when using $p \cdot n_{\mathrm{ID}}$ to approximate $|D|$.

Since the expected number of drawn white marbles is an estimation for $|H(M) \setminus D|$, we can set $E[X] = \hat{\vartheta}_{|H(M)\setminus D|}$, where

$$\hat{\vartheta}_{|H(M)\setminus D|} = \hat{\vartheta}_{|H(M)\cup D|} - p \cdot n_{\mathrm{ID}}$$

is used to estimate $|H(M) \setminus D| = |H(M) \cup D| - |D|$.

Solving the resulting formula for $|H(M)|$ yields our estimator $\hat{\vartheta}_{|M|}$ for $|M|$:

$$\hat{\vartheta}_{|M|} = \frac{\hat{\vartheta}_{|H(M)\setminus D|}}{(1-p)} = \frac{\hat{\vartheta}_{|H(M)\cup D|} - p \cdot n_{\mathrm{ID}}}{(1-p)}.$$

Combining the estimates for $\hat{\vartheta}_{|M|}$, $\hat{\vartheta}_{|M\setminus D|}$ and $\hat{\vartheta}_{|M\cup D|}$ (see Equation (1)) finally yields

$$\hat{\vartheta}_{|M|} = \frac{n_{\mathrm{ID}} \cdot (k - p \cdot \max(K))}{(1-p) \cdot \max(K)}. \tag{4}$$

This provides a way to estimate set cardinalities for single P2KMV sketches. We now extend this to set operations for P2KMV—first unions, then intersections.

### 4.3 Set Union Cardinality Estimation

Perturbation has an impact on combining multiple sketches as well. For sketches without dummies ($p = 0$), all hash values in $K_u$ are by definition elements of the union. As seen before, estimating $S_u$'s cardinality is, in this case, straightforward (see Section 3.4). For P2KMV sketches ($p > 0$), there is a chance, though, that a hash value in $K_u$ does not correspond to an ID sampled by *any* sketch, i. e., it is actually not in the union.

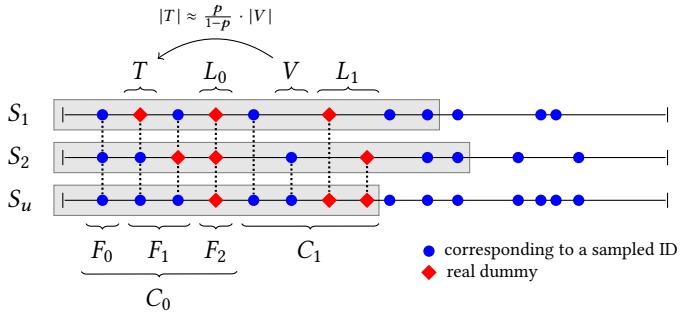



Fig. 3. Illustrating $F_i$, $C_i$, and $L_i$, which are relevant for the intersection cardinality estimation. The relation between the subsets $V$ and $T$ serves as an example for our main idea of removing the real dummies' influence.

However, we can use the perturbation probabilities $p_i$ of the P2KMV sketches $S_i$ to calculate $S_u$'s perturbation probability $p_u$. We assume that the dummies are chosen independently for the individual sketches. Let $p_u$ denote the probability that a specific ID was chosen as a dummy in at least one of the $S_1, S_2, \ldots, S_n$ sketches. Then, $p_u$ equals

$$p_u = 1 - \prod_{i=1}^{n} (1 - p_i).$$

Assuming all $S_i$ use the same privacy level $p$, $p_u$ is simply given by $1 - (1-p)^n$.

The set union cardinality of perturbed P2KMV sketches can thus be estimated by evaluating the union sketch $S_u$ according to Equation (4) using $p_u$ as the privacy level.

### 4.4 Set Intersection Cardinality Estimation

As we have seen before, set intersection cardinality estimation is a challenge even without perturbation. Adding perturbation makes estimations even more challenging. Without special consideration, the dummy elements would cause massive estimation errors. So, we finally come to the main question tackled in this work: how to obtain unbiased intersection estimates from P2KMV sketches? We start with an intuitive explanation with two sets, which shows the general approach. In particular, we show how to infer the Jaccard index in the presence of perturbation through dummy elements. Subsequently, we, more formally, consider the general case of $n$ sets. Like in the case of set union cardinality estimations it would be possible to formulate all equations for individual per-sketch privacy levels. However, as this would greatly reduce the possibility to simplify the resulting equations, we only consider set intersection cardinality estimations using sketches with identical privacy levels.

*4.4.1 The case $n = 2$.* Let $M_1$ and $M_2$ be two sets, $S_1$ and $S_2$ the corresponding P2KMV sketches with (identical) privacy level $p$, and $S_u$ the respective union sketch holding $k_u$ hash values. Further let $H(M_1)$ and $H(M_2)$ be the sets of hash values corresponding to the IDs in $M_1$ and $M_2$. Recall, our goal for estimating $|M_1 \cap M_2|$ efficiently is to find an estimate $\hat{\vartheta}_{|U|}$ of the union cardinality and an estimate $\hat{\vartheta}_J$ of the Jaccard index.

To this end, let $\mathcal{I}$ be the (finite) set of possible IDs. Let $D_i \subseteq \{1, 2, \ldots, n_{\mathrm{ID}}\}$ be the (unknown) subset of all hash values that were chosen as dummy elements for $S_i$ and let $RD_i = D_i \setminus H(M_i)$ be the set of "real" dummies for $S_i$, i. e., all dummy hash values that do not correspond to an ID in $M_i$. Finally, by $K_i$ we denote the set of $S_i$'s $k_i$ smallest hash values.

In order to estimate the Jaccard index, the first step is to determine $|C_0|$, the number of hash values that are in $K_1, K_2$, *and* $K_u$, which is given by

$$C_0 = \{h \in K_u : h \in (H(M_1) \cup RD_1) \cap (H(M_2) \cup RD_2)\}\,.$$

Since P2KMV sketches are obfuscated by dummy elements, however, $|C_0|$ per se can merely be used to give an upper-bound estimation of the true intersection size. For our two-set example, we can identify four cases, illustrated in Figure 3, that have an impact on $C_0$: (a) $h$ corresponds to an ID $x$ that has been sampled by $S_1$ and $S_2$ (●,●), (b) $h = H(x)$ is a real dummy in $S_1$ but $x$ has been sampled by $S_2$ (◆,●), (c) $h$ corresponds to an ID $x$ that has been sampled by $S_1$ but $H(x)$ is a real dummy in $S_2$ (●,◆), and (d) $h$ is a real dummy in $S_1$ and $S_2$ (◆,◆).

Let $F_i$ denote the set of hash values that are real dummies in exactly $i$ sketches and correspond to sampled IDs in the remaining sketches. We show the four cases and the respective sets $F_i$ in the lower part of Figure 3. We can now decompose $c_0 = |C_0|$ by

$$c_0 = |F_0| + |F_1| + |F_2|. \tag{5}$$

For the Jaccard index estimation, we need to find the ratio between hash values that correspond to sampled IDs in *all* P2KMV sketches and the number of hash values in $K_u$ corresponding to sampled IDs in *any* of the P2KMV sketches. Here the hash values in $K_u$ corresponding to sampled IDs in *any* of the P2KMV sketches, i. e. (●,●), (●,◆/-), and (◆/-,●)—where ◆/- translates to real dummy *or* none —are representatives of the union. The hash values that were sampled by all P2KMV sketches, i. e. (●,●), however, represent elements in the intersection. Consequently, we need to determine $|F_0|$, i. e., the number of hash values that correspond to IDs sampled by every sketch (●,●). The perturbation prevents us from individually distinguishing hash values of sampled IDs from dummies, though.

However, we can still estimate the set size $|F_0|$. To this end, we start from Equation (5). In order to obtain $|F_0|$ from this equation, we need additional information. Instead of only considering the hash values that are present in both P2KMV sketches ($h \in K_1$ *and* $h \in K_2$, i. e., $h \in C_0$), we also take into account the hash values that are only in exactly *one* P2KMV sketch (*either* $h \in K_1$ *or* $h \in K_2$).

Let $C_i$ with $i \in \{0, 1, \ldots, n-1\}$ be the hash values in $K_u$ that are present in exactly $n - i$ P2KMV sketches, either as a hash value of a sampled ID or as a real dummy, and let $c_i = |C_i|$. For two sets, only $C_0$ and $C_1$ are defined. $C_0$, containing (◆/●, ◆/●) as used before, already satisfies this definition. $C_1$, containing (◆/●, -) and (-, ◆/●), provides the additional information necessary to estimate the cardinality of $F_0$. The composition of both sets is depicted in Figure 3.

We now show how $C_1$ helps to estimate $|F_1|$ and $|F_2|$. To this end, we define two subsets $V$ and $T$ as an example to explain the general approach. Let $V$ consists of all $h \in K_u$ that were neither real dummies nor hashes of sampled IDs in $S_1$, i. e., $h \notin K_1$, but corresponding to IDs sampled by $S_2$ (-,●). Let $T$, in contrast, consists of all hash values in $K_u$ that were real dummies in $S_1$ and correspond to sampled IDs in $S_2$ (◆,●). The upper part of Figure 3 illustrates the relation between $V$ and $T$.

One possible approach to understand the connection between $V$ and $T$ is the following: Due to the initial perturbation, about a fraction of $p$ of all hash values that are not hash values of IDs sampled by $S_1$, i. e. $h \notin H(M_1)$, will be chosen as dummy values resulting in real dummies, i. e. $h \in RD_1$. All hash values neither chosen as dummies nor corresponding to a sampled ID will be absent from $S_1$ and thus $h \notin K_1$. Thus, $V$'s and $T$'s cardinalities can be approximated by

$$\begin{aligned} |V| &= |\{h \in K_u : h \notin K_1 \wedge h \in H(M_2)\}| \\ &\approx (1-p) \cdot |\{h \in K_u : h \notin H(M_1) \wedge h \in H(M_2)\}|, \end{aligned}$$

$$\begin{aligned} |T| &= |\{h \in K_u : h \in RD_1 \wedge h \in H(M_2)\}| \\ &\approx p \cdot |\{h \in K_u : h \notin H(M_1) \wedge h \in H(M_2)\}|. \end{aligned}$$

The cardinalities of $T$ and $V$ are therefore related (as indicated in Figure 3) by

$$|T| \approx \frac{p}{1-p} \cdot |V|.$$

Following this line of thought, $|F_1|$ can be estimated by

$$\hat{\vartheta}_{|F_1|} = \frac{p}{1-p} \cdot (c_1 - |L_1|), \tag{6}$$

where $L_i$ is the set of hash values in $K_u$ that are real dummies in exactly $n-i$ sketches but absent in the remaining sketches. For two sets there are exactly two $L_i$: $L_0$ containing (♦,♦) and $L_1$ containing (-,♦) as well as (♦,-). Since

$$|L_0| = |\{h \in K_u : h \in RD_1 \wedge h \in RD_2\}|,$$

$$\begin{aligned} |L_1| = &|\{h \in K_u : h \in RD_1 \wedge h \notin K_2\}| \\ &+ |\{h \in K_u : h \notin K_1 \wedge h \in RD_2\}|, \end{aligned}$$

the expected value of their cardinalities $E\left[X_{|L_0|}\right]$ and $E\left[X_{|L_1|}\right]$ can be written as

$$E\left[X_{|L_0|}\right] = p^2 \cdot |\{h \in K_u : h \notin H(M_1) \wedge h \notin H(M_2)\}|,$$

$$E\left[X_{|L_1|}\right] = 2p \cdot (1-p) \cdot |\{h \in K_u : h \notin H(M_1) \wedge h \notin H(M_2)\}|.$$

Given an estimate $\hat{\vartheta}_{|L_0|}$ of $|L_0|$, we can estimate $|L_1|$ by

$$\hat{\vartheta}_{|L_1|} = 2\hat{\vartheta}_{|L_0|} \cdot \frac{1-p}{p}. \tag{7}$$

We can obtain an estimate for $|L_0|$ by considering $d$, the number of real dummies in $K_u$. For two sets, $h \in \{1, 2, \ldots, n_{\mathrm{ID}}\}$ is a real dummy in $S_u$ iff $h \in L_0$ or $h \in L_1$, i. e., $d = |L_0| + |L_1|$. Thus, we can estimate $\hat{\vartheta}_{|L_0|}$ using Equation (7) by

$$\hat{\vartheta}_{|L_0|} = \frac{\hat{\vartheta}_d}{1 + 2 \cdot \frac{1-p}{p}}.$$

Apart from estimating the size of other $L_i$, $\hat{\vartheta}_{|L_0|}$ can be used to estimate $|F_2|$. For two sets, $F_2$ contains all $h \in K_u$ that are real dummies in both sketches (♦,♦). Hence, $L_0 = F_2$ holds. The only missing component $\hat{\vartheta}_d$ (the estimate for the total number of real dummies in $K_u$) can be derived easily as follows.

Note that no hash value corresponding to a sampled ID in $S_u$ can be a real dummy. Each of the remaining hash values is chosen with probability $p_u$ as a dummy. Since every dummy thus chosen will be a real dummy we can now estimate $\hat{\vartheta}_{|RD|}$ the number of IDs chosen as dummies for $S_u$ that are real dummies. To this end, we will use the expected number of hash values chosen as dummies among all hash values that are not in $H(M_u)$, i. e. do not correspond to IDs sampled by $S_u$. Using the set cardinality estimation for the union $\hat{\vartheta}_{|U|}$ as in Equation (4) we can express $\hat{\vartheta}_{|RD|}$ by

$$\hat{\vartheta}_{|RD|} = p_u \cdot \left(n_{\mathrm{ID}} - \hat{\vartheta}_{|U|}\right).$$

Now we can estimate $R_{RD}$, the number of $S_u$'s real dummies in proportion to $|H(M_1) \cup H(M_2) \cup D_u|$, by

$$R_{RD} = \frac{\hat{\vartheta}_{|RD|}}{\hat{\vartheta}_{|RD|} + \hat{\vartheta}_{|U|}}.$$

This finally yields an estimate $\hat{\vartheta}_d$ via the expected number of real dummies in $K_u$ given $R_{RD}$:

$$\hat{\vartheta}_d = k_u \cdot R_{RD}.$$

Since we can derive $c_0$ and $c_1$ directly from $S_1$ and $S_2$, we can use $\hat{\vartheta}_{|L_0|}$ to estimate $|F_2|$ and $|L_1|$ (see Equation (7)), which allows us to estimate $|F_1|$ (see Equation (6)) and finally $|F_0|$ (see Equation (5)).

To estimate the Jaccard index, we determine the number of hash values in $K_u$ that correspond to sampled IDs in *any* sketch, which is given by

$$k_u - \hat{\vartheta}_d = k_u \cdot (1 - R_{RD}).$$

Bringing it all together, we can estimate $\hat{\vartheta}_J$ the Jaccard index of $M_1$ and $M_2$ as

$$\hat{\vartheta}_J = \frac{\hat{\vartheta}_{|F_0|}}{k_u \cdot (1 - R_{RD})}, \tag{8}$$

and the cardinality of the set intersection of $M_1$ and $M_2$ by

$$\hat{\vartheta}_{|M_1 \cap M_2|} = \hat{\vartheta}_J \cdot \hat{\vartheta}_{|U|}. \tag{9}$$

Following the described approach, we can obtain estimates for $|F_i|$ based on $c_j$ for an arbitrary number of sets.

*4.4.2 Generalization to n sets.* The approach introduced above can be extended to allow the estimation of set intersection cardinalities for $n$ sets. Let $S_1, S_2, \ldots, S_n$ be these sets' P2KMV sketches with privacy level $p$. We will continue to use $F_i$, $c_j$, $d$, and $L_t$ as defined above with $i \in \{0, 1, \ldots, n\}$ and $j, t \in \{0, 1, \ldots, n-1\}$. Observe that for $n$ sets, $n$ different $c_j$ can be obtained from $S_1, S_2, \ldots, S_n$.

While $c_j$ and $R_{RD}$ can be calculated as before, the remaining equations have to be adapted for $n$ sets. Estimating $|L_t|$ is still possible using the expected value of $|L_0|$. To this end, one can make use of the fact that every $L_t$ is a union of $\binom{n}{t}$ disjoint subsets. These only differ in which sets their elements are real dummies in. Hence, $|L_t|$'s expected value $E\left[X_{|L_t|}\right]$ can be calculated by

$$E\left[X_{|L_t|}\right] = \left(\frac{1-p}{p}\right)^t \cdot \binom{n}{t} \cdot E\left[X_{|L_0|}\right].$$

Estimating $\hat{\vartheta}_d$ as before, we can estimate $|L_0|$ by

$$\hat{\vartheta}_{|L_0|} = \frac{\hat{\vartheta}_d}{\sum_{t=0}^{n-1} \left(\frac{1-p}{p}\right)^t \cdot \binom{n}{t}} = \frac{\hat{\vartheta}_d}{\left(\frac{1}{p}\right)^n - \left(\frac{1-p}{p}\right)^n}, \tag{10}$$

and $|L_t|$ using

$$\hat{\vartheta}_{|L_t|} = \left(\frac{1-p}{p}\right)^t \cdot \binom{n}{t} \cdot \frac{\hat{\vartheta}_d}{\left(\frac{1}{p}\right)^n - \left(\frac{1-p}{p}\right)^n}.$$

Further, for $n$ sets $|F_n| = |L_0|$ holds, enabling us to estimate $|F_n|$ using $\hat{\vartheta}_{|L_0|}$.

In general, the connection between $c_j$ and $F_i$ is more intricate. First we note, $c_0$ remains unchanged:

$$c_0 = \sum_{i=0}^{n} |F_i|. \tag{11}$$

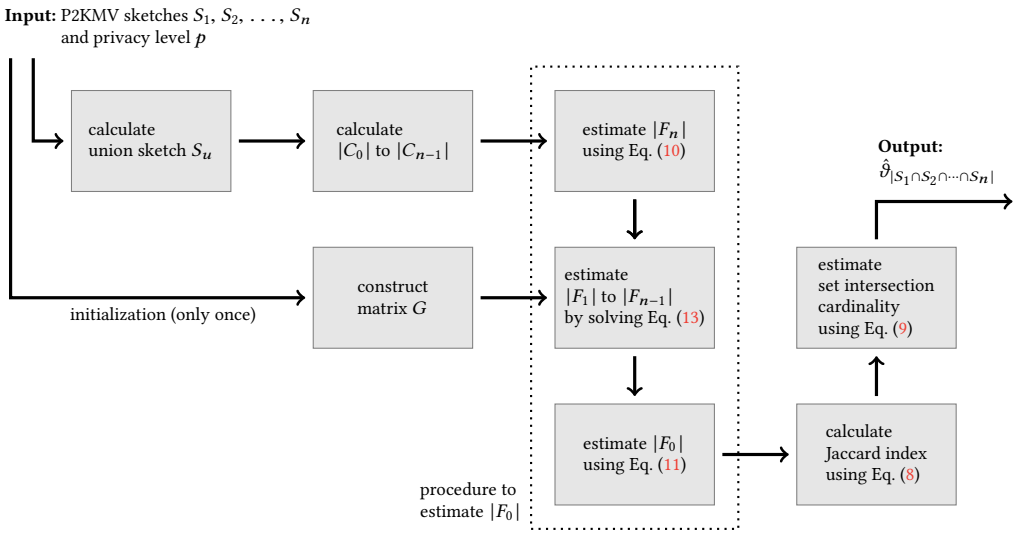



Fig. 4. Estimating intersection cardinality for $n$ sets.

By following the approach introduced for $n = 2$, a general formula for the connection between $c_j$ and $|F_i|$ can be obtained (for details see Appendix A):

$$c_j \left(\frac{p}{1-p}\right)^j - \binom{n}{j} \cdot \frac{\hat{\vartheta}_d}{\left(\frac{1}{p}\right)^n - \left(\frac{1-p}{p}\right)^n} = \sum_{k=1}^{n-j} \binom{n-k}{j} |F_{n-k}|. \tag{12}$$

Let $G \in \mathbb{R}^{(n-1)\times(n-1)}$ be the matrix with coefficients

$$g_{i,j} = \begin{cases} \binom{n-j}{i-j} & \text{, if } i \geq j \\ 0 & \text{, else.} \end{cases}$$

Then $f = (|F_1|, |F_2|, \ldots, |F_{n-1}|)^T$ can be estimated by solving the following system of linear equations:

$$G \cdot f = \begin{pmatrix} c_{n-1} \left(\frac{p}{1-p}\right)^{n-1} - \binom{n}{n-1} \cdot \frac{\hat{\vartheta}_d}{(\frac{1}{p})^n - (\frac{1-p}{p})^n} \\ \vdots \\ c_1 \left(\frac{p}{1-p}\right)^1 - \binom{n}{1} \cdot \frac{\hat{\vartheta}_d}{(\frac{1}{p})^n - (\frac{1-p}{p})^n} \end{pmatrix}. \tag{13}$$

Because now estimates for $|F_1|, |F_2|, \ldots, |F_n|$ are known, $|F_0|$ can be estimated using $c_0$ and Equation (11). This allows the estimation of the Jaccard index using Equation (8), which in turn yields an estimate for the set intersections cardinality.

Figure 4 concludes this section and summarizes the necessary steps to estimate the set intersection cardinality for $n$ P2KMV sketches.

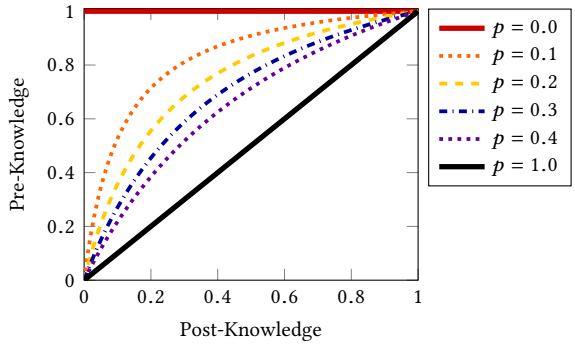



Fig. 5. Influence of the privacy level on the adversary's post-knowledge, showing that standard KMV ($p = 0$) cannot protect the privacy of all users. P2KMV ($p > 0$) instead effectively limits the knowledge gain and provides plausible deniability.

## 5 EVALUATION

In this section, we evaluate P2KMV and provide an in-depth parameter study. In particular, we take a look on the influence of various parameters on the privacy, accuracy, and on the runtime complexity. Moreover, we provide guidelines on parameter selection and, finally, discuss our findings and P2KMV's limitations.

### 5.1 Privacy Analysis

We have introduced the notion of pre- and post-knowledge, which provides a well-defined and formally tractable basis for analysis. We also elaborated on this understanding and prooved plausible deiniability for our approach. From Theorem 4.1, it becomes clear that if we choose a privacy level $p > 0$, we gain plausible deniability. That is, an uncertainty remains and the adversary's post-knowledge will effectively be limited.

In Figure 5, we plot the attacker's change in knowledge for varying privacy levels. We compare the adversary's pre-knowledge ($x$ axis) to the adversary's post-knowledge ($y$ axis). For $p = 0$ the adversary is able to link the $k$ smallest hash values with certainty to an ID. For $p = 1$ the adversary is unable to learn any new information. At the same time, though, we cannot extract any useful statistics either. Thus, a privacy level between these two extremes is desirable.

Considering that an increase in the privacy level reduces the accuracy of subsequent estimations, a use-case specific sweet spot between privacy and accuracy has to be found. In the following, we will therefore focus on the accuracy.

### 5.2 Methodology

We have implemented a deterministic simulation to evaluate the accuracy our approach. In our simulation environment, we know the ground truth of sampled IDs, and are therefore able to assess the achieved accuracy.

To this end, we generated (pseudo) random sets that follow a discrete uniform distribution on the integers in $[1, n_{\mathrm{ID}}]$. We paid particular attention to the set generation, which we designed to yield a controlled set intersection size: we first randomly generated the intersection and used it as a basis for each set. In a second step, we complemented the individual sets with distinct random values, as long as they did not expand the set intersection. This way of set generation gives us total

control of the resulting set intersection cardinality, but otherwise imposes little restrictions on the resulting sets.

For P2KMV, the generated random numbers in the set were directly used as hash values, because they already fulfilled our specification of $H$. Each simulation was repeated ten times independently by changing the random seed, which resulted in different sets and perturbation patterns.

In our evaluation, we compare our approach to PCSA [19] and to a Bloom filter-based approach [30]. PCSA stands for probabilistic counting with stochastic averaging and is based on a so-called FM sketch: values are hashed into binary sketches. The resulting bit-pattern serves as an indicator for the number of distinct values. The estimate is then improved by using stochastic averaging, which combines $m$ trials into a better estimate. Estimating set intersection cardinalities with PCSA is possible using the inclusion-exclusion principle.

Bloom filters can also be used to estimate set intersection cardinalities: every set is represented by a Bloom filter $B_1, B_2, \ldots, B_n$. These Bloom filters are combined into a new filter $B_\wedge$ by calculating the bit-wise logical AND of $B_1, B_2, \ldots, B_n$. Because of the nature of Bloom filters, $B_\wedge$ may contain 1-bits that do not belong to any element in the set intersection. Many et al. [30] provide a methodology to correct these false positives, and Papapetrou et al. [36] provide a cardinality estimation. We refer the reader to Many et al.'s paper for more details.

Unlike for privacy-enhanced PCSA [40] and P2KMV, to our best knowledge, there is no Bloom filter-based approach that provides equivalent privacy guarantees. We therefore focus on PCSA and P2KMV when it comes to comparing privacy-preserving approaches.

In fact, both, privacy-enhanced PCSA and P2KMV, have comparable privacy properties when choosing the parameters correctly. When we look at the hash values that could be linked to IDs *without* perturbation, an adversaries' worst case knowledge-gain is identical for P2KMV and privacy-enhanced PCSA, if both use the same privacy level $p$. For privacy-enhanced PCSA about one ID per row in the PCSA matrix needs to be protected through obfuscation. Therefore, privacy-enhanced PCSA with $k$ rows and P2KMV (storing $k$ hash values) will have identical worst case privacy properties, if they use the same privacy level $p$. Thus, the results are directly comparable.

We, therefore, choose our simulation parameters accordingly: in order to demonstrate the applicability in large settings, we chose a simulation scenario with $n_{\mathrm{ID}} = 10^7$ IDs and individual sets with a cardinality $s$ of $2^{19}$ IDs, but varying intersection sizes. If not specified otherwise, we use seven sets and $k = 0.01 \cdot s = 5243$, which equals one percent of the total set size. We use the same number $k$ for the number of rows in a PCSA matrix. By default, we also use the same privacy level, i. e., $p = 0.1$, for both approaches, PCSA and P2KMV. In all our evaluation results, we capped negative estimates to zero, since negative cardinalities do not make sense. We show the arithmetic means of ten simulation runs. Error bars indicate the standard error of the mean.

### 5.3 Baseline Accuracy ($p = 0$)

Here, we provide a baseline evaluation, i. e., without any additional privacy protection. We compare our approach to PCSA using the inclusion-exclusion principle and to the Bloom filter-based approach mentioned before.

Each of the three approaches has one or more parameters, which greatly influence their accuracy. In order to make the three different approaches comparable, we choose the parameters to yield a similar worst-case privacy, i. e., approximately the same number of IDs that can be revealed from the data structure.

In Figure 6, we compare the accuracy for varying intersection cardinalities. The ground truth cardinality is plotted along the $x$ axis, the estimated set intersection cardinality along the $y$ axis. We informally refer to the *resolution limit* as the lower bound where the respective approach starts

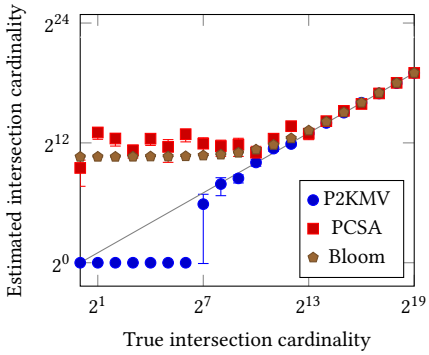



Fig. 6. Baseline accuracy ($k = 0.01 \cdot s$, $p = 0$, 7 sets).

to provide accurate results. For PCSA and Bloom filter the resolution limit is approximately $2^{14}$ and $2^{12}$ elements, respectively. In contrast, our P2KMV approach has a resolution limit of $2^8$, i. e., it already yields accurate results for intersection sizes that are significantly smaller. Note that the error bars, showing the standard error of the mean, appear to have a downward overhang due to the logarithmic scale of the plot. In fact, they are symmetric around the mean.

The data points prior to the resolution limit provide a deeper insight into the respective approaches and are therefore noteworthy as well. The quality of the set intersection cardinality estimation with P2KMV depends on a representative number of elements in the intersection being hashed to the sketch's smallest values. For very small intersection sizes, there is a high probability that no elements of the intersection are among the $k$ smallest hash values. Thus, due to the limited number of samples, the resulting estimate is too small. For PCSA and the Bloom filter, in contrast, the estimation procedure is quite noisy and starts producing accurate results when the signal (the set intersection cardinality) is significantly stronger than this noise (the inherent variance of the estimation process).

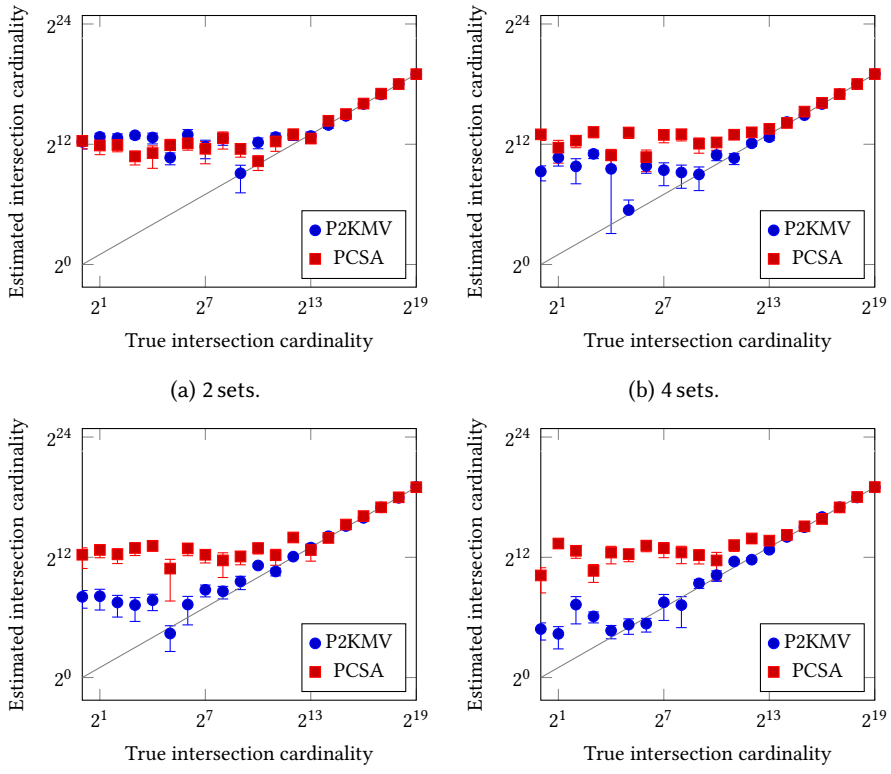
### 5.4 Accuracy for $p > 0$

Now, let us turn to the accuracy of perturbed sketches. Since there is no Bloom filter-based approach that provides equivalent privacy guarantees, we focus on privacy-enhanced PCSA and P2KMV in the remainder.

In Figure 7, we start by examining how the number of sets influences accuracy. The figures show simulation results for varying true intersection cardinalities ($x$ axis) and compare it to the estimated set intersection cardinalities ($y$ axis). Again, note the logarithmic scale and the resulting downward overhang of the error bars. The gray diagonal provides a guideline: perfect estimates would lie here.

Starting by an intersection of two sets and gradually increasing the number of sets, the results clearly demonstrate the benefits of P2KMV over PCSA. While P2KMV's resolution limit improves with an increasing number of sets, PCSA's resolution limit gets worse. It appears that the influence of the privacy level becomes less noticeable for P2KMV. This underlines P2KMV's merits to handle larger number of sets. The variations of the resolution limit are basically caused by an accumulated privacy level $p_u$, which is larger for more sets. PCSA's resolution limit, in contrast, suffers not only from an accumulated $p_u$, but also from error propagation.

Increasing the privacy level $p$ intentionally adds more noise to the sketches. Therefore, it inevitably affects estimations. In Figure 8, we show $p$'s impact on accuracy. While increasing $p$




(a) 2 sets.

(b) 4 sets.

(c) 6 sets.

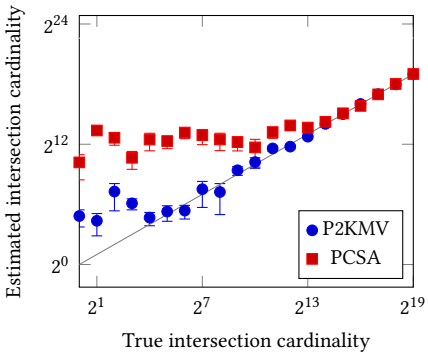
(d) 8 sets.

Fig. 7. Influence of #sets ($k = 0.01 \cdot s$, $p = 0.1$).

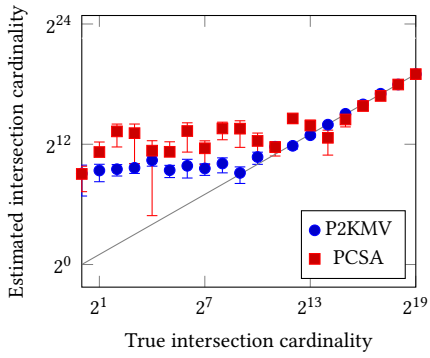
massively distorts PCSA, P2KMV can handle it adequately, resulting only in a shifted resolution limit.

We can adjust the accuracy by modifying $k$. In Figure 9, we show $k$'s influence on the accuracy. The results, including Figure 8a ($k = 0.01 \cdot s$), confirm that increasing $k$ results in better accuracy for both PCSA and P2KMV. For P2KMV, $k$'s impact is stronger, though. The transition from Figure 9a to 8a and from Figure 8a to 9b clearly demonstrates P2KMV's accuracy improvements.

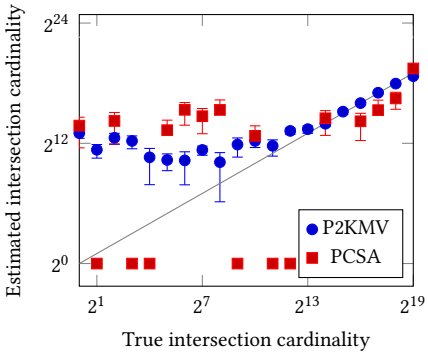
So far, we presented the standard error of the mean in our analysis to provide a measure of accuracy. In order to quantify the amount of variation or dispersion as well, we also provide the standard deviation for varying setups. In Table 1, we fixed the true intersection size at $2^{14}$ and varied the number of sets, $p$, and $k$ as before. Hence, the results can be considered a point estimate of our simulation study. In some cases, the standard deviation becomes larger, e. g., for less sets or a larger $p$, which implies a dispersion between individual runs. As a consequence, we suggest to perform multiple measurements to increase precision (see Section 5.7 for a discussion). In general, though, the standard deviation confirms our impression of the parameter's influence and that P2KMV is superior compared to PCSA in terms of intersections.



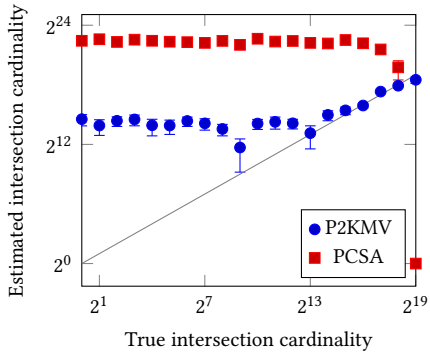

(a) $p = 0.1$.




(b) $p = 0.2$.

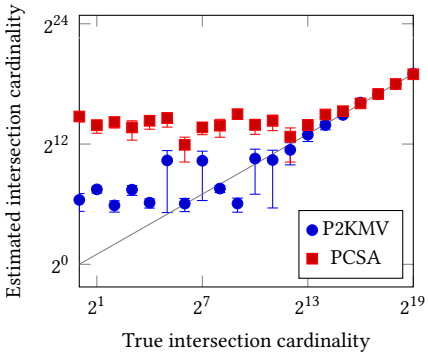



(c) $p = 0.3$.

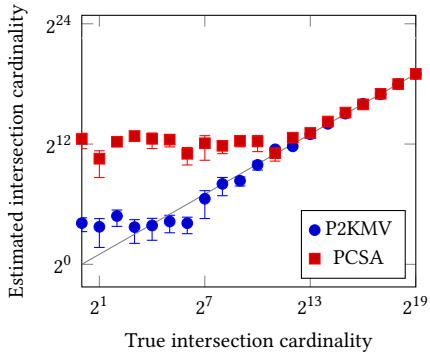



(d) $p = 0.4$.

Fig. 8. Influence of $p$ ($k = 0.01 \cdot s$, 7 sets).




(a) $k = 0.001 \cdot s$.




(b) $k = 0.02 \cdot s$.

Fig. 9. Influence of $k$ ($p = 0.1$, 7 sets).

Table 1. Set intersection cardinality estimations for a true intersection size of $2^{14}$ (= 16384) IDs.

| | P2KMV | | PCSA | |
|---|---|---|---|---|
| Setup | Mean | SD | Mean | SD |
| 7 sets, $p = 0.0$, $k = 0.01 \cdot s$ | 16381 | 2477 | 18504 | 9547 |
| 7 sets, $p = 0.1$, $k = 0.01 \cdot s$ | 16779 | 4293 | 19338 | 18493 |
| 7 sets, $p = 0.1$, $k = 0.02 \cdot s$ | 16587 | 2960 | 19408 | 7002 |
| 7 sets, $p = 0.3$, $k = 0.01 \cdot s$ | 15859 | 9193 | 23132 | 50788 |
| 2 sets, $p = 0.1$, $k = 0.01 \cdot s$ | 15539 | 10283 | 20742 | 12732 |

SD = standard deviation

### 5.5 Parameter Selection for P2KMV

As discussed above, there are several parameters influencing P2KMV's accuracy and privacy.

Both are influenced by the choice of the privacy level $p$. In general we advice to use a privacy level of 0.1, because we believe it provides a good trade-off between privacy and accuracy. However, higher estimation accuracy can be achieved at the cost of weaker plausible deniability by a privacy level closer to zero.

The number $k$ of the smallest values stored in each sketch greatly influences the estimate's accuracy as well as how many IDs have to be secured via perturbation. As we saw in Figure 9, the choice of $k$ is directly related to the sketch's resolution limit. To better understand the impact of $k$ on P2KMV's accuracy, taking a closer look at KMV can be very helpful. It is important to note that both accuracy and resolution limit of KMV represent lower bounds for P2KMV—in essence because the perturbation can obviously only make things worse. Because of the way the set intersection cardinality estimation is performed (see Equation (2)), $k$ influences the estimate as follows: given the union cardinality $|U|$, $k$ determines the "granularity" of the estimation, because it can only output multiples of $\frac{|U|}{k}$ as estimates. This also gives a theoretical lower bound for the resolution limit, as no set intersection cardinalities below $\frac{|U|}{k}$ can be told apart.

As stated before, these lower bounds for KMV are also lower bounds for P2KMV and can be used to suitably choose $k$ for a given application. Now let us assume somebody wants to know whether the cardinality of $n$ sets' intersection is bigger than $i$. This person knows that the intersected sets are each of size $s$. Then $k$ should chosen to be at least

$$\frac{n \cdot s}{i},$$

which is the maximal cardinality of the sets' union divided by the desired intersection cardinality. Our practical experience showed that a more robust estimate can be obtained by choosing $k$ about twice as big, i. e.,

$$k \geq 2 \cdot \frac{n \cdot s}{i}.$$

Note that this approximates the union cardinality rather coarsely, which results in substantial overestimation if $n$ is large. If the cardinality $u$ of the union is known or can be approximated, replacing $n \cdot s$ with $u$ in this last inequality will yield a far better choice for $k$.

Table 2. Mean runtime for varying number of sets

| | P2KMV | | PCSA | |
|---|---|---|---|---|
| #Sets | Mean | SE | Mean | SE |
| 9 | 1228 ms | 102 ms | 1102 ms | 171 ms |
| 10 | 1331 ms | 144 ms | 2108 ms | 162 ms |
| 11 | 1455 ms | 139 ms | 4907 ms | 269 ms |
| 12 | 1384 ms | 139 ms | 9399 ms | 667 ms |
| 13 | 1546 ms | 165 ms | 20140 ms | 344 ms |

SE = standard error of the mean

### 5.6 Runtime Complexity

Beside accuracy, computational efficiency is one of our main motivations for the development of P2KMV. The runtime for estimating the intersection cardinality using the inclusion-exclusion principle grows exponentially with the number of sets $n$, as shown above. P2KMV, in contrast, is far more efficient: to estimate the set intersection cardinality with P2KMV, the first step is to construct matrix $G$, which takes $O(n^2)$ operations. Solving the system of linear equations from Equation (13) efficiently involves calculating the inverse of $G$. This takes $O(n^3)$ steps. However, both constructing $G$ and calculating its inverse has to be done only once for any given value of $n$. So, when calculating multiple set intersections of $n$ sets each, the matrix can be pre-calculated and reused.

As the second step, the union sketch $S_u$ has to be calculated, which is necessary for many procedures in our algorithm. In order to calculate $S_u$, we have to find the $k_u$ smallest hash values in the (sorted) P2KMV sketches $S_1, S_2, \ldots, S_n$, which results in $O(k_u \cdot n)$ operations.

The key parameters for set intersection cardinality estimations are the coefficients $c_j$. We can obtain them by examining how often which hash value appeared in $S_1, S_2, \ldots, S_n$. This will take $O(k_{max} \cdot n)$ steps, where $k_{max}$ denotes the highest number of stored hash values in the P2KMV sketches.

From the coefficients $c_j$, we can estimate the $|F_i|$. Estimating $|F_n|$ using Equation (10) requires the set cardinality of the union. This can be estimated with the help of $S_u$ by finding the biggest hash value of the $k_u$ hash values in $S_u$ and performing a constant number of multiplications, divisions and subtractions. We recommend to store the hash values sorted by size, which enables us to perform the set cardinality estimation in $O(1)$.

In order to estimate $|F_1|$ to $|F_{n-1}|$, Equation (13) is solved. Here the inverse of $G$ simplifies this procedure to a standard matrix-vector-multiplication, which takes $O(n^2)$ operations. With the help of $|F_1|$ to $|F_n|$, $|F_0|$, our cardinality of interest, is calculated by subtracting $c_0$ by $|F_1|$ to $|F_n|$. This last step takes $O(n)$ operations, resulting in a runtime complexity of $O(n^2)$ for estimating the $|F_i|$.

Finally the Jaccard index can be calculated by evaluating Equation (8). Each value needed here was already calculated in one of the previous steps and no complex math is involved, resulting in a complexity of $O(1)$. The same applies for estimating the set intersection cardinality using the Jaccard index, which is in $O(1)$, too.

The overall runtime complexity of our estimation algorithm depends on whether $G$ and its inverse were already calculated for an earlier estimation. If so, the complexity of estimating the set intersection cardinality using P2KMV is $O(n^2)$, because $k$ will always be smaller or equal to $n$ in P2KMV. Otherwise, $O(n^3)$ operations are necessary to calculate $G^{-1}$. In summary, the asymptotic

complexity of Jaccard index-based estimation is drastically lower than the asymptotic effort that results from any solution based on the inclusion-exclusion principle. It is important to note that the privacy level does not affect the runtime complexity of P2KMV.

To verify our theoretical runtime analysis, we experimentally evaluated PCSA and P2KMV by measuring the time to calculate the set intersection cardinality. Here we used a testbed with an Intel Xeon E5-2643 v2 3.5 GHz, 8 GB RAM, Ubuntu Linux 14.04, and Java 1.7. Table 2 shows the mean runtime for 9 to 13 sets, measured in ten independent experiments. Even within this short range of relatively small numbers of sets, we see that PCSA rapidly becomes very expensive. P2KMV on the other hand, achieves good results for any number of sets, which underlines our theoretical findings.

### 5.7 Discussion

Despite complementing the toolbox of privacy-preserving data structures in general and privacy-preserving counting sketches in particular, P2KMV has some limitations, which we will discuss in this section.

First, we stress that our focus is clearly on efficient and accurate set intersection cardinality estimations. For general cardinality estimations and for union cardinality estimations other counting sketches exist, which achieve better accuracy. For example, PCSA was developed to estimate the set cardinality in *large* data streams. Therefore, it does not come as a surprise that PCSA's cardinality estimation is more accurate than P2KMV. Nevertheless, P2KMV still provides reasonable results. Our previous results implicitly include regular cardinality estimations and therefore also underline the accuracy of P2KMV in this use case.

In order to achieve this high accuracy with P2KMV, we suggest parameter tuning as discussed before. In addition, we suggest to repeat the same measurement to increase the confidence in the statistical results. Likewise, splitting the user base into cohorts within the same measurement window might be able to increase confidence as well. Since the memory footprint and the computational complexity of P2KMV is reasonably low, we can afford such approaches.

With P2KMV's data minimization approach we achieve a high degree of privacy for the vast majority of users. By applying our perturbation technique, we can guarantee a certain degree of plausible deniability even to the small group of users that are amongst the $k$ samples. For many use cases, this kind of plausible deniability probably suffices, but there might be situations where also the absence of users reveals sensitive information. In future work, we envisage to extend our approach to conceal the absence of users as well. This is not in the nature of counting sketches, though. It requires a novel "symmetric" perturbation technique, which also removes elements from the sketch, and therefore leads to a completely different estimation formula.

In the face of a very powerful adversary with considerable knowledge about a certain user ID and its statistical characteristics, a very interesting property is revealed. We can assume that such an adversary knows the user's hash value and the respective sketches it would expect this user's hash value to be sampled. In this highly targeted attack, if the user is found in all sketches, the plausible deniability is less convincing. To some extent, our notion of an adversary's pre-knowledge covers this attack vector. That is, the adversary already has a "suspicion", i. e., assumes with a high probability, that the user contacted the service. Relative to this high starting point, the certainty that and adversary gains, i. e., its post-knowledge after inspecting the sketches, is considerably small. While we expect in practice a large number of categorical variables measured with counting sketches, which likely leave enough uncertainty towards an adversary, the attack vector still remains relevant and requires investigation in the future. In combination with the idea of symmetric perturbation, we could impede this kind of attack vector.

In summary, we were able to show P2KMV's high accuracy in a deterministic simulation. We showed that it improves privacy and provides provable plausible deniability. Further, we verified its efficiency both theoretically and practically.

## 6 RELATED WORK

In this paper, we investigate methods to obtain privacy-preserving user statistics. To this end, we introduced a way to estimate set intersection cardinalities in a privacy-preserving way. The majority of approaches in the area of privacy-preserving statistics consider calculating set intersections or set intersection cardinalities in a distributed setting. Instead, we consider a centralized setting, which is very common in practice. In this section, we will give a brief overview of the design space and discuss related approaches. We put a focus on the security against an external and state-level adversary and on the economic consequences of proposed solutions.

### 6.1 Distributed Approaches

There is a wide range of publications on private set intersections or private set intersection cardinality based on distributed computation [10, 28, 30]. They generally follow two principles: (a) user data is stored by distributed entities, and (b) only the computation result is made public, i. e., nobody can learn any further information about other private input sets. Most prominent for this group of solutions are algorithms employing two-party computation and secure multi-party computation.

*6.1.1 Two-Party Computation.* The oldest of these classes of algorithms is two-party computation, which can be seen as the predecessor of multi-party computation. Two-party computation was first introduced by Yao [44] and allows *two* users to privately evaluate a function, i. e., only the function's result is made public but not the users' input.

A common approach to two-party computation is the use of *garbled circuits*, e. g. [4, 23, 42]. However, to compute set intersections or their cardinalities using two-party computation, there are many alternatives to garbled circuits like the use of oblivious transfer [37], homomorphic encryption [12, 25], or commutative encryption [1, 8, 11].

All two-party computation algorithms have in common that they compute set intersections of two parties only, which severely limits their applicability for association rule mining, for example. For some of the approaches [5, 29] there are corresponding multi-party protocols overcoming this limitation, while others [25] are strictly limited to two sets.

*6.1.2 Secure Multi-Party Computation.* Secure multi-party computation (MPC) describes a group of algorithms to privately compute a function, i. e., without revealing any input data but what can be learned by the function's public output. Here, more than two private data sets can be used and calculating the intersection of many private sets becomes possible.

As before, there are some generic MPC approaches to compute set intersections or set intersection cardinalities, e. g., using garbled circuits [5], or specialized solutions. The main approach for specialized solutions seems to be the usage of either secret sharing [14, 30, 34], homomorphic encryption [26, 28], commutative encryption [41] or distributed differential privacy [10, 33].

While a wide variety of very clever algorithms has been developed in this field of research, a severe drawback regarding privacy remains. In MPC either all data is stored in the system—by the individual data sources or as secret shares distributed over all parties—or data minimization techniques are used in a way that does not guarantee protection of each data record. So when all parties of the multi-party computation are breached or are forced to reveal their information by a state level adversary some or all private information remain retrievable.

### 6.2 Centralized Approaches

Instead of collecting user data on many different sources, in the centralized approach all data is collected at one central location (the service provider) and has to be secured there. This model not only reflects today's practice, but also eliminates the need of user cooperation. In fact, the centralized model covers situations where data accumulates implicitly, i. e., metadata, which the user cannot control. For this reason, users have to trust the service provider to some extent anyway.

This trust, however, does not extend to attackers who might gain access to the centralized data in one way or the other. While certain attacks might be deflected by good data encryption, others will also steal or otherwise obtain the secrets used for encryption. For this reason, storing all data and performing set intersection or set intersection cardinality computations on the plain data, which is arguably the easiest way, poses a privacy risk for the users. To mitigate this privacy risk and to allow for the computation of set intersection cardinalities either sketches or differential privacy can be used.

*6.2.1 Sketches.* The purpose of sketches is to aggregate data in a way that retains important characteristics of the data, and reduces its storage footprint. This makes them suited as tools for data minimization as well. Furthermore, the aggregation can result in an increase in privacy for certain users, if their data can not be recovered sufficiently from the aggregate.

In general, sketches are used in a wide variety of ways [9, 13, 24, 27]. Counting sketches were first devised by Flajolet and Martin [19] to enable fast and memory efficient estimates of the number of unique elements in a database. Usually, they achieve a small memory footprint by constructing a succinct representation of a set [3, 19, 21]. Sketches are also often used to reduce communication complexity in network protocols [13, 30, 38]. Here, we use sketches to reduce the amount of stored personal data of individuals. While in the first case, sketches might still contain sensible information, we aim to quantify and minimize the information leakage remaining. Our proposed algorithm therefore provides further procedures to obfuscate the data.

The combination of counting sketches and privacy-enhancing obfuscation was first introduced in [40]. In this paper, we follow similar lines, but use KMV [3] as the counting sketch instead of PCSA [20]. Changing the underlying counting sketch requires novel approaches to sketch evaluation, but, as we have shown in this paper, this pays off through efficient and accurate set-intersection cardinality estimates. The role of the perturbation in our new counting sketch is to prevent the attacker from fully recovering user data from the aggregate. This increases the users' privacy and guarantees plausible deniability for *every* user.

*6.2.2 Differential Privacy.* In recent years, differential privacy [16] became a popular technique for obfuscating the answers to statistical queries [15, 31]. Similar to our approach, this obfuscation, which can be seen as the addition of random noise, allows for strong privacy guarantees. Because the user's privacy has to be protected against an attacker with access to the centrally stored data, it is not enough to apply the random noise to computation results, as is normally done [10, 33]. Instead, the noise has to be applied to the underlying data sets directly.

Another notable approach is RAPPOR [17], a privacy-preserving technology to crowdsourcing statistics. It uses a randomized response scheme [43] to achieve differential privacy, which requires the clients' assistance. Metadata, such as a software version for example, are often an inherent part of a protocol, though. Unfortunately, randomized response in general and RAPPOR in particular cannot protect privacy in these cases because altering these values might impair the protocol. Metadata, however, are relevant sources for user statistics and therefore needs additional protection applied by service provider as well. Since it is not obvious how to implement RAPPOR in a centralized

fashion, particularly how to deal with data that accumulates over time, we believe that our approach complements the set of technologies for privacy-preserving user statistics.

In general, unlike perturbation, which obfuscates the presence of a user in a set, differential privacy obfuscates the user's absence as well. Thus, the resulting data set would be heavily distorted by noise, which is bound to severely limit the accuracy of any computations based on them. To our best knowledge, there is no publication showing how accurate set intersection cardinality estimates can be achieved when using data sets that are differentially private with regard to membership of elements in the set.

## 7 CONCLUSION

In this paper, we have shown that counting sketches are a promising basis to achieve data minimization. Without additional means of protection, though, they still leak personal information. Therefore, we have developed P2KMV, a privacy-preserving counting sketch based on KMV.

As the main contribution of this paper, we showed that P2KMV is particularly suited to calculate set intersection cardinalities. To this end, we modeled P2KMV stochastically and developed the required tools to estimate cardinalities. While our approach is efficient and accurate, our model also demonstrates privacy guarantees, i. e., plausible deniability.

In our evaluation, we reveal parameter dependencies and thereby confirm the findings of our formal approach. In summary, we believe P2KMV finds a sweet spot in the trade-off between privacy, accuracy, and efficiency.

## A GENERALIZING $c_j$ AND $|F_i|$

A central contribution of this paper is the derivation of a perturbation-resistant estimation formula for the set intersection cardinality. While the estimation for two sets can be derived more or less straightforward, the generalization for $n$ sets is much more abstract and complex. To this end, the connection between $c_j$ and $|F_i|$, given in Equation (12), plays a central role for our estimation. Therefore, we will present an in-depth derivation of their connection in this section.

Let $M_1, M_2, \ldots, M_n$ be the $n$ sets whose set intersection cardinality is calculated. Further let $H(M_1), H(M_2), \ldots, H(M_n)$ be their hash values. To simplify the notation for $n$ sets, let $\pi_1, \pi_2, \ldots, \pi_{n!}$ denote all permutations for the numbers 1 to $n$. Accordingly, we can write $F_i$, in a compact form, as

$$F_i = \bigcup_{t=1}^{n!} \left\{ h \in K_u : \bigwedge_{l_1=1}^{i} h \in RD_{\pi_t(l_1)} \wedge \bigwedge_{l_2=i+1}^{n} h \in H(M_{\pi_t(l_2)}) \right\}.$$

Further, we can write the expected value of $F_i$'s cardinality $X_{|F_i|}$ as

$$E\left[X_{|F_i|}\right] = p^i \cdot \left| \bigcup_{t=1}^{n!} \left\{ h \in K_u : \bigwedge_{l_1=1}^{i} h \notin H(M_{\pi_t(l_1)}) \wedge \bigwedge_{l_2=i+1}^{n} x \in H(M_{\pi_t(l_2)}) \right\} \right|.$$

Similarly, we can describe $C_i$ as

$$\begin{aligned} C_i &= \bigcup_{t=1}^{n!} \left\{ h \in K_u : \bigwedge_{l_1=1}^{i} h \notin K_{\pi_t(l_1)} \wedge \bigwedge_{l_2=i+1}^{n} h \in K_{\pi_t(l_2)} \right\} \\ &= \bigcup_{t=1}^{n!} \left\{ h \in K_u : \bigwedge_{l_1=1}^{i} h \notin K_{\pi_t(l_1)} \wedge \bigwedge_{l_2=i+1}^{n} \left(h \in H(M_{\pi_t(l_2)}) \vee h \in RD_{\pi_t(l_2)}\right) \right\}. \end{aligned}$$

Since $h \in H(M_i)$ and $x \in RD_i$ are mutually exclusive, we can partition $C_i$ into a union of $n-i+1$ disjoint subsets $Q_0, Q_1, \ldots, Q_{n-i}$, where

$$Q_m = \bigcup_{t=1}^{n!} \left\{ h \in K_u : \bigwedge_{l_1=1}^{i} h \notin K_{\pi_t(l_1)} \wedge \bigwedge_{l_2=i+1}^{i+m} h \in H(M_{\pi_t(l_2)}) \wedge \bigwedge_{l_3=i+m+1}^{n} h \in RD_{\pi_t(l_3)} \right\}.$$

Our cardinality estimation is based on the fact that the cardinality's expected value of the sets $\Theta_1 = \{h \in K_u : h \notin K_i\}$ and $\Theta_2 = \{h \in K_u : h \in RD_i\}$ are connected. Let $X_{|S_1|}$ and $X_{|S_2|}$ be the random variables of their respective cardinality, then

$$E\left[X_{|\Theta_1|}\right] = (1-p) \cdot |\{h \in K_u : h \notin H(M_i)\}|$$

and

$$E\left[X_{|\Theta_2|}\right] = p \cdot |\{h \in K_u : h \notin H(M_i)\}|$$

holds.

We use this fact to write the expected value of $Q_m$'s cardinality $X_{|Q_m|}$ as

$$\begin{aligned} E\left[X_{|Q_m|}\right] &= (1-p)^i \cdot p^{n-i-m} \cdot \alpha(m) \cdot \\ &\quad \left| \bigcup_{t=1}^{n!} \left\{ h \in K_u : \bigwedge_{l_1=1}^{i} h \notin H(M_{\pi_t(l_1)}) \wedge \bigwedge_{l_2=i+1}^{i+m} h \in H(M_{\pi_t(l_2)}) \wedge \bigwedge_{l_3=i+m+1}^{n} h \notin H(M_{\pi_t(l_3)}) \right\} \right| \\ &= (1-p)^i \cdot p^{n-i-m} \cdot \alpha(m) \cdot \frac{E\left[X_{|F_{n-m}|}\right]}{p^{n-m}} \\ &= \frac{(1-p)^i}{p^i} \cdot \alpha(m) \cdot E\left[X_{|F_{n-m}|}\right]. \end{aligned}$$

Here, $\alpha(m)$ is introduced to enumerate the number of *different* sets that constitute $Q_m$ but have the *same* cardinality estimation. For example the expected cardinality of the sets

$$\Theta_3 = \left\{ h \in K_u : \bigwedge_{l_1=1}^{i} h \notin K_{l_1} \wedge \bigwedge_{l_2=i+1}^{i+m} h \in H(M_{l_2}) \wedge \bigwedge_{l_3=i+m+1}^{n} h \in RD_{l_3} \right\}$$

and

$$\Theta_4 = \left\{ h \in K_u : h \in RD_1 \wedge \bigwedge_{l_1=2}^{i} h \notin K_{l_1} \wedge \bigwedge_{l_2=i+1}^{i+m} h \in H(M_{l_2}) \wedge \bigwedge_{l_3=i+m+1}^{n-1} h \in RD_{l_3} \wedge h \notin K_n \right\}$$

is

$$(1-p)^i \cdot p^{n-i-m} \cdot \left| \left\{ h \in K_u : \bigwedge_{l_1=1}^{i} h \notin H(M_{l_1}) \wedge \bigwedge_{l_2=i+1}^{i+m} h \in H(M_{l_2}) \wedge \bigwedge_{l_3=i+m+1}^{n} h \notin H(M_{l_3}) \right\} \right|.$$

So the expected value of $X_{|S_3 \cup S_4|}$, that is, the union's cardinality of $S_3$ and $S_4$, would be

$$E\left[X_{|S_3 \cup S_4|}\right] = 2 \cdot (1-p)^i \cdot p^{n-i-m} \cdot \left| \left\{ h \in K_u : \bigwedge_{l_1=1}^{i} h \notin H(M_{l_1}) \wedge \bigwedge_{l_2=i+1}^{i+m} h \in H(M_{l_2}) \wedge \bigwedge_{l_3=i+m+1}^{n} h \notin H(M_{l_3}) \right\} \right|.$$

The number of distinct sets $\alpha(m)$, which share the same formula for their expected cardinality, can therefore simply be described by

$$\alpha(m) = \binom{n-m}{i}.$$

Thus, the complete formula for the expected cardinality of $Q_m$ is given by

$$E\left[X_{|Q_m|}\right] = \frac{(1-p)^i}{p^i} \cdot \binom{n-m}{i} \cdot E\left[X_{|F_{n-m}|}\right].$$

This cardinality estimation can be used to express the expected cardinality of $C_i$ by

$$\begin{aligned}
E\left[X_{|C_i|}\right] &= |\bigcup_{m=0}^{n-i} Q_m| \\
&= \sum_{m=0}^{n-i} |Q_m| \\
&= \sum_{m=0}^{n-i} \frac{(1-p)^i}{p^i} \cdot \binom{n-m}{i} \cdot E\left[X_{|F_{n-m}|}\right] \\
&= \frac{(1-p)^i}{p^i} \cdot \sum_{m=0}^{n-i} \binom{n-m}{i} \cdot E\left[X_{|F_{n-m}|}\right] \\
&= \frac{(1-p)^i}{p^i} \cdot \left(\binom{n}{i} \cdot E\left[X_{|F_n|}\right] + \sum_{m=1}^{n-i} \binom{n-m}{i} \cdot E\left[X_{|F_{n-m}|}\right]\right) \\
&= \frac{(1-p)^i}{p^i} \cdot \left(\binom{n}{i} \cdot \frac{\hat{\vartheta}_d}{\left(\frac{1}{p}\right)^n - \left(\frac{1-p}{p}\right)^n} + \sum_{m=1}^{n-i} \binom{n-m}{i} \cdot E\left[X_{|F_{n-m}|}\right]\right).
\end{aligned}$$

Replacing $E\left[X_{|C_i|}\right]$ with $c_i$ and rearranging the equation for the sum of the $F_i$ then gives the formula presented in Equation (12).

## REFERENCES


[1] Rakesh Agrawal, Alexandre V. Evfimievski, and Ramakrishnan Srikant. 2003. Information Sharing Across Private Databases. In *SIGMOD '03: Proceedings of the ACM SIGMOD International Conference on Management of Data*. 86–97.

[2] Ero Balsa, Carmela Troncoso, and Claudia Díaz. 2012. OB-PWS: Obfuscation-Based Private Web Search. In *IEEE Symposium on Security and Privacy*. IEEE Computer Society, 491–505.

[3] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, D. Sivakumar, and Luca Trevisan. 2002. Counting Distinct Elements in a Data Stream. In *RANDOM '02: Randomization and Approximation Techniques, 6th International Workshop*.

[4] Donald Beaver, Silvio Micali, and Phillip Rogaway. 1990. The Round Complexity of Secure Protocols (Extended Abstract). In *STOC '90: Proceedings of the 22th Annual ACM Symposium on Theory of Computing*. 503–513.

[5] Assaf Ben-David, Noam Nisan, and Benny Pinkas. 2007. FairplayMP: a system for secure multi-party computation. In *CCS '08: Proceedings of the 15th ACM Conference on Computer and Communications Security*. 257–266.

[6] Kevin Beyer, Peter J. Haas, Berthold Reinwald, Yannis Sismanis, and Rainer Gemulla. 2007. On Synopses for Distinct-value Estimation Under Multiset Operations. In *SIGMOD '07: Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*.

[7] Vincent Bindschaedler, Reza Shokri, and Carl A. Gunter. 2017. Plausible Deniability for Privacy-Preserving Data Synthesis. *PVLDB* 10, 5 (2017), 481–492.

[8] Carlo Blundo, Emiliano De Cristofaro, and Paolo Gasti. 2014. EsPRESSO: Efficient privacy-preserving evaluation of sample set similarity. *Journal of Computer Security* 22, 3 (2014), 355–381.

[9] Haowen Chan, Adrian Perrig, Bartosz Przydatek, and Dawn Xiaodong Song. 2007. SIA: Secure information aggregation in sensor networks. *Journal of Computer Security* 15, 1 (2007), 69–102.

[10] Ruichuan Chen, Alexey Reznichenko, Paul Francis, and Johannes Gehrke. 2012. Towards Statistical Queries over Distributed Private User Data. In *NSDI '12: Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation*. 169–182.

[11] Emiliano De Cristofaro, Paolo Gasti, and Gene Tsudik. 2012. Fast and Private Computation of Cardinality of Set Intersection and Union. In *CANS '12: Proceedings of the 11th International Conference on Cryptology and Network Security*.

[12] Dana Dachman-Soled, Tal Malkin, Mariana Raykova, and Moti Yung. 2009. Efficient Robust Private Set Intersection. In *ACNS '09: Proceedings of the 7th International Conference on Applied Cryptography and Network Security*.

[13] Stefan Dietzel, Andreas Peter, and Frank Kargl. 2015. Secure Cluster-Based In-Network Information Aggregation for Vehicular Networks. In *VTC '15-Spring: Proceedings of the 81st IEEE Vehicular Technology Conference*.

[14] Changyu Dong, Liqun Chen, and Zikai Wen. 2013. When private set intersection meets big data: an efficient and scalable protocol. In *CCS '13: Proceedings of the 20th ACM Conference on Computer and Communications Security*.

[15] Marlon Dumas, Luciano García-Bañuelos, and Peeter Laud. 2016. Differential Privacy Analysis of Data Processing Workflows. In *Graphical Models for Security - Third International Workshop, GraMSec 2016, Lisbon, Portugal, June 27, 2016, Revised Selected Papers*. 62–79.

[16] Cynthia Dwork. 2006. Differential Privacy. In *ICALP '06-2: Automata, Languages and Programming, 33rd International Colloquium*. 1–12.

[17] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *CCS '14: Proceedings of the 21st ACM Conference on Computer and Communications Security*. 1054–1067.

[18] Theodore G Faticoni. 2013. *Combinatorics : an introduction / Theodore G. Faticoni.* Wiley, Hoboken, NJ.

[19] Philippe Flajolet and G. Nigel Martin. 1983. Probabilistic Counting. In *FOCS '83: 24th Annual Symposium on Foundations of Computer Science*.

[20] Philippe Flajolet and G. Nigel Martin. 1985. Probabilistic Counting Algorithms for Data Base Applications. *J. Comput. Syst. Sci.* 31, 2 (1985), 182–209.

[21] Stefan Heule, Marc Nunkesser, and Alex Hall. 2013. HyperLogLog in Practice: Algorithmic Engineering of a State of The Art Cardinality Estimation Algorithm. In *EDBT '13: Proceedings of the joint 2013 EDBT/ICDT Conferences*.

[22] Jochen Hipp, Ulrich Güntzer, and Gholamreza Nakhaeizadeh. 2000. Algorithms for Association Rule Mining - A General Survey and Comparison. *SIGKDD Explorations* 2, 1 (2000), 58–64.

[23] Yan Huang, David Evans, Jonathan Katz, and Lior Malka. 2011. Faster Secure Two-Party Computation Using Garbled Circuits. In *20th USENIX Security Symposium, San Francisco, CA, USA, August 8-12, 2011, Proceedings*. USENIX Association.

[24] Michael Kamp, Christine Kopp, Michael Mock, Mario Boley, and Michael May. 2013. Privacy-Preserving Mobility Monitoring Using Sketches of Stationary Sensor Readings. In *ECMLPKDD '13: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*.

[25] Murat Kantarcioglu, Robert Nix, and Jaideep Vaidya. 2009. An Efficient Approximate Protocol for Privacy-Preserving Association Rule Mining. In *PAKDD '09: Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*. 515–524.

[26] Lea Kissner and Dawn Xiaodong Song. 2005. Privacy-Preserving Set Operations. In *CRYPTO '05: Proceedings of the 25th Annual International Cryptology Conference*.

[27] Peter Lieven and Björn Scheuermann. 2010. High-Speed Per-Flow Traffic Measurement with Probabilistic Multiplicity Counting. In *INFOCOM '10: Proceedings of the 29th Annual Joint Conference of the IEEE Computer and Communications Societies*.

[28] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. 2012. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *STOC '12: Proceedings of the 44th ACM Symposium on Theory of Computing Conference*.

[29] Dahlia Malkhi, Noam Nisan, Benny Pinkas, and Yaron Sella. 2004. Fairplay - Secure Two-Party Computation System. In *USENIX Security '04: Proceedings of the 13th USENIX Security Symposium*. 287–302.

[30] Dilip Many, Martin Burkhart, and Xenofontas Dimitropoulos. 2012. *Fast Private Set Operations with SEPIA*. Technical Report TIK report no. 345. ETH Zurich.

[31] Frank McSherry and Ratul Mahajan. 2010. Differentially-private network trace analysis. In *SIGCOMM '10: Proceedings of the 2010 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*.

[32] Luca Melis, George Danezis, and Emiliano De Cristofaro. 2015. Efficient Private Statistics with Succinct Sketches. *CoRR* abs/1508.06110 (2015). http://arxiv.org/abs/1508.06110

[33] Arjun Narayan and Andreas Haeberlen. 2012. DJoin: Differentially Private Join Queries over Distributed Databases. In *OSDI '12: Proceedings of the 10th USENIX Symposium on Operating Systems Design and Implementation*.

[34] G. Sathya Narayanan, T. Aishwarya, Anugrah Agrawal, Arpita Patra, Ashish Choudhary, and C. Pandu Rangan. 2009. Multi Party Distributed Private Matching, Set Disjointness and Cardinality of Set Intersection with Information Theoretic Security. In *CANS '09: Proceedings of the 8th International Conference on Cryptology and Network Security*.

[35] Open Whisper Systems. 2016. Grand jury subpoena for Signal user data, Eastern District of Virginia. https://whispersystems.org/bigbrother/eastern-virginia-grand-jury/. (Oct. 2016).
[36] Odysseas Papapetrou, Wolf Siberski, and Wolfgang Nejdl. 2010. Cardinality estimation and dynamic length adaptation for Bloom filters. *Distributed and Parallel Databases* 28, 2-3 (2010), 119–156.
[37] Benny Pinkas, Thomas Schneider, and Michael Zohner. 2014. Faster Private Set Intersection Based on OT Extension. In *USENIX Security '14: Proceedings of the 23rd USENIX Security Symposium*.
[38] Alex Rousskov and Duane Wessels. 1998. Cache Digests. *Computer Networks* 30, 22-23 (1998), 2155–2168.
[39] Adi Shamir. 1979. How to Share a Secret. *Commun. ACM* 22, 11 (1979), 612–613.
[40] Florian Tschorsch and Björn Scheuermann. 2013. An algorithm for privacy-preserving distributed user statistics. *Computer Networks* 57, 14 (2013), 2775–2787.
[41] Jaideep Vaidya and Chris Clifton. 2005. Secure set intersection cardinality with application to association rule mining. *Journal of Computer Security* 13, 4 (2005), 593–622.
[42] Justin Wagner, Joseph N. Paulson, Xiao Wang, Bobby Bhattacharjee, and Héctor Corrada Bravo. 2016. Privacy-preserving microbiome analysis using secure computation. *Bioinformatics* 32, 12 (2016), 1873–1879.
[43] Stanley L. Warner. 1965. Randomized response: A survey technique for eliminating evasive answer bias. *J. Amer. Statist. Assoc.* 60, 309 (1965), 63–69.
[44] Andrew Chi-Chih Yao. 1982. Protocols for Secure Computations (Extended Abstract). In *FOCS '82: 23rd Annual Symposium on Foundations of Computer Science*. 160–164.