PRISM: Private Retrieval of the Internet's Sensitive Metadata

Ang Chen
University of Pennsylvania

Andreas Haeberlen
University of Pennsylvania

Abstract

The Internet is producing a wealth of data about its own operation, in the form of NetFlow records, routing table entries, traffic statistics, etc. Several previous works – including, for instance, Clark's "knowledge plane" – have considered the idea of building a giant distributed database that (at least conceptually) contains all of this information. Such a database could have many attractive uses, including distributed troubleshooting, attack mitigation, or traffic management. However, so far the idea has not been realized, and it is likely that privacy concerns have played a role.

In this paper, we ask whether *differential privacy* could provide the strong privacy guarantees that would be needed to put this idea into practice. We discuss some key concerns that have been raised about differential privacy, such as its limited scalability and its finite "privacy budget", and we point out several characteristics of the Internet that could mitigate these concerns. We also sketch the design of PRISM, a system for differentially private queries on NetFlow records that could form the basis of a potential "knowledge plane".

1 Introduction

The proposal to create a "knowledge plane" for the Internet has been around for almost a decade – it goes back to a paper by Clark et al. [24] – and the network is producing a wealth of data that could be used for this purpose, e.g., in the form of NetFlow records, routing tables, or data from a variety of active and passive measurement systems [43, 44, 37]. Numerous papers have shown the advantages of a collaborative analysis, which could rely on a "global view" of the Internet instead of merely data that each network collects locally; also, a variety of potential applications have been suggested, including collaborative network measurement [58, 60, 59], distributed troubleshooting [17, 16], forecasting [9], cooperative intrusion detection [70], botnet analysis [56, 10], and many others (e.g., [66]).

However, despite the wealth of data and the many possible applications, such a "knowledge plane" has not yet become a reality. We postulate that this is, at least in part, due to privacy concerns. As recent events have shown, seemingly innocent metadata, such as IP addresses, can be used to infer very personal information. Consider, for instance, the example of Paula Broadwell and former CIA director David Petraeus [64]: even though the two used an anonymous email account to communicate, Broadwell was eventually tracked down by the FBI by correlating the IP addresses she had logged in from with the hotels at which she had been staying. Other well-publicized cases, such as the Netflix prize [12] and the AOL search data [11], have demonstrated that even a good-faith effort to protect privacy, e.g., by anonymizing or "scrubbing" data before release, cannot reliably prevent a privacy disaster. Thus, it is not surprising that ISPs are reluctant to make data available, or sometimes even attempt to deliberately hide information [39, 7, 25].

In this paper, we ask whether *differential privacy* [29] could be the enabling technology for an Internet knowledge plane. Differential privacy is one of the strongest forms of privacy available; it essentially promises to each individual I that, when a query is answered about the data, the answer would have been almost as likely if I's data had not been included. Thus, differential privacy is very well suited for releasing large trends in the data (such as the high-level traffic flow across a network) while effectively protecting the privacy of individuals. Most importantly, differential privacy rests on solid mathematical foundations: it can be formally proven that accidental privacy leaks, like the ones experienced by Netflix and AOL, are impossible – even under very pessimistic assumptions.

We discuss several common concerns about differential privacy, and we examine whether they would apply to a possible knowledge plane. For instance, differential privacy is often discussed in terms of a centralized database that contains all the private information (which, in the Internet, would be a privacy nightmare in itself!),

and it has the concept of a finite "privacy budget" that must be charged for the "privacy cost" of queries and that, once exhausted, prevents further queries forever. We find that it should be possible to address these concerns: briefly, the centralized database could be replaced by a network of ISP-local databases that answer queries using a distributed cryptographic protocol, and the budget, while certainly finite, is enormous and could probably be replenished slowly over time; thus, it may be possible to keep answering queries indefinitely, without jeopardizing privacy.

We also sketch the design of a system called PRISM that enables Private Retrieval of the Internet's Sensitive Metadata. Unlike its namesake that has received so much media attention [5], *PRISM would allow access to data only with very strict differential privacy guarantees*. For concreteness, we discuss PRISM in the context of NetFlow data, but we hypothesize that it could be extended to other types of data that exists on the Internet, such as routing tables or access logs.

Differential privacy is not a magic bullet – there are interesting and useful queries that cannot be answered with such strong privacy guarantees. For instance, we might legitimately want to identify the command node of a botnet, or to obtain a list of nodes that are infected by a particular type of malware. These queries identify specific individuals, which is the very thing differential privacy is designed to prevent! However, PRISM does not have to be the *only* way that queries can be answered – human administrators can still approve and answer any query they wish, just as it is done today. PRISM would *add* a way to answer certain "safe" queries automatically, in cases where very strong privacy assurances can be given. Thus, PRISM could enable ISPs to safely obtain at least some of the potential benefits that have been suggested for a knowledge plane.

2 Overview

In this section, we briefly review the key definitions of differential privacy, and then sketch a possible architecture for the PRISM system.

2.1 Background: Differential privacy

Differential privacy [29] is a property of randomized queries that take a database as input and return a result that is typically some form of aggregate (e.g., a histogram, or a count of items that have some property of interest). The database is viewed as a collection of rows, and each row contains data about one individual. Intuitively, differential privacy promises that each row has only a very small impact on the overall result of the query. More formally, a query q (with range R) is ε-differentially private if, for all databases B and B' that differ in at most one row, and for all possible outputs $S \subseteq R$,

$$Pr[q(B) \in S] \leq e^{\varepsilon} \cdot Pr[q(B') \in S] \quad (1)$$

In other words, if we add or remove a single individual's data, the probability that the (randomized) output will fall into some set S can change by at most a factor of e^{ε}. Here, ε is a privacy parameter; smaller values of ε yield stronger privacy.

If q is a numerical query – say, a count – a common way to achieve differential privacy is to use the *Laplace mechanism* [30]. Suppose $\bar{q}$ is the precise query (without the noise). Then the Laplace mechanism first computes the precise result $\bar{q}(B)$, based on the data in the database B, and then adds noise from a Laplace distribution – i.e., returns $q(B) := \bar{q}(B) + \mathrm{Lap}(\lambda)$. The parameter λ of the Laplace distribution, which controls the amount of noise that is to be added, depends on the *sensitivity* of the query: If $|\bar{q}(B) - \bar{q}(B')| \leq s$ for any pair of databases B, B' that differ in at most one row, then the sensitivity of $\bar{q}$ is s, and the parameter λ is chosen to be $\lambda = s/\varepsilon$.

When q is a non-numerical query – e.g., one that returns elements from a set, such as AS numbers – the Laplace mechanism is not appropriate. However, there are other mechanisms that can answer such queries, e.g., the exponential mechanism [48].

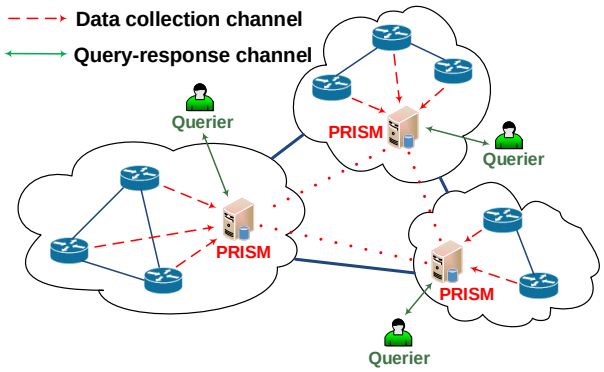

Figure 1: Proposed architecture of PRISM. Each ISP operates a node that has access to its local data and participates in a distributed query protocol. The private data itself never leaves the ISP.

2.2 PRISM architecture

In order to serve as a useful "knowledge plane", we would like PRISM to have access to as much data as possible, including sensitive data – such as NetFlow records – that would not normally be available for querying. At

the same time, we insist on strong, provable privacy guarantees for the individuals (the ISPs' customers) whose data is accessed through PRISM. We obviously cannot prevent ISPs from accessing their own local data, but we can ensure that the sensitive data never leaves the ISP that collected it, *except* through differentially private queries.

Figure 1 shows a proposed architecture that achieves these goals. Each ISP operates a local PRISM node that is given unrestricted access to local information; however, the PRISM node (unlike its namesake) only answers queries that it can certify as differentially private, e.g., through static analysis [33]. Only other ISPs are authorized to ask queries, and each PRISM node maintains a "privacy budget" to limit the amount of private information that is revealed as more and more queries are answered over time. In Section 4, we discuss the privacy budget in more detail.

Not all queries can be answered by querying ISPs individually; for instance, to reveal a botnet's command-and-control structure, it may be necessary to combine data from several ISPs [56]. PRISM can support queries that join data from multiple ISPs, and it can answer them using a secure, distributed query processing protocol, such as DJoin [52]. Even in this case, private data does not have to leave the network of the ISP that collected it: DJoin uses cryptographic techniques such as private set intersection [45, 31] to ensure that only differentially private results can be observed outside of its domain.

3 Case study: NetFlow

We hope that many different types of data will eventually be available through PRISM; however, for concreteness, we focus on NetFlow records for the purposes of this paper. Cisco's NetFlow [23] and its variants (e.g., Sampled NetFlow [6], IPFIX [15]) provide a well-standardized solution for collecting flow-level measurements, and they are already widely supported by many vendors [4]. They comprise an important data source both for research [53, 41] and for industrial innovations [2, 3]. In fact, sharing NetFlow traces has already been proposed [8], but without differential privacy guarantees.

In this context, PRISM would provide the abstraction of a global database that contains NetFlow records from the entire Internet. The "rows" of this database would contain the flows to and from a specific IP address. (Recall from Section 2.1 that differential privacy protects the privacy of rows.) While we would ideally like the rows to have all the data that pertain to a specific *individual*, this seems impractical because there is no way to tell which individual(s) caused a given packet to be sent. IPs seem like a reasonable approximation. In other words, PRISM aims to answer queries on NetFlow data without revealing too much about which IPs fit the query criteria.

3.1 Example queries

For concreteness, we now give a few examples of queries that PRISM might support.

Easy queries: The "easiest" queries for PRISM are queries that can be broken into subqueries, such that each subquery can be answered by an individual ISP. This includes common queries, e.g., counts, that serve as a basis for many advanced analyses [21]: spikes in the number of flows, e.g., may indicate port scanning, flash crowds, etc [40]. A query might ask PRISM to count the number of flows exceeding a certain size, e.g., containing more than 100 packets and lasting for more than 60 seconds [50], because extreme counts may trigger a subsequent DDoS detection. In an SQL-like syntax, the query could be written as follows:

```
SELECT COUNT(f.id) FROM ISP 1-N
WHERE (f.pkts>100 AND f.duration>60)
```

A variety of techniques have been proposed that could answer such queries, including [21, 55]; similar techniques are available for querying, e.g., histograms [21] or aggregate time-series [63]. If some of these techniques are included in PRISM, answering queries such as the example above is clearly feasible.

Harder queries: If a query cannot be broken into per-ISP subqueries, it is more difficult for PRISM to answer, but by no means impossible. For instance, a querier might try to trace attack flows back to their source using a sequence of cross-domain joins; thus, we could gain a backtrace capability without a specialized infrastructure for that purpose, such as [42]. For instance, a querier could expose the source ASes of spoofed traffic in the entire Internet by checking whether a flow's source IP address is contained in the ISP it originates from:

```
SELECT f.SrcASN
FROM JOIN Internet BY FlowID
WHERE (f.SrcIP ∉ f.OrigISP)
```

Or, we could change the WHERE predicate to obtain the source ASes of all traffic ending in darknets [68]:

```
WHERE (f.DstIP is unallocated)
```

PRISM could rely on DJoin [52] for answering queries like this. Since the answer in this case is a set and not a number, the Laplace mechanism is not appropriate (noising an AS number does not make sense), but PRISM could use the exponential mechanism [48] instead. So queries of this type are more difficult (and potentially more computationally expensive) but still seem feasible.

Hard queries: There is no doubt that there are some interesting queries that we currently do not know how to answer efficiently. For instance, we might want to use a "similarity join" query [19], which joins databases

by key similarity instead of exact match. For example, a similarity join based on the similarity of flow on/off patterns could allow queriers to find correlated Internet flows, which could help to expose stepping stones [71]. Right now, we do not know how to support similarity joins, but, as research progresses, PRISM could be extended with more advanced query processing techniques.

4 The Privacy Budget

A common concern about differential privacy is that it can only answer a finite number of queries. In this section, we examine how severe this concern would be in the context of an Internet knowledge plane.

4.1 The problem

Recall the guarantee from Section 2.1: if an ε-differentially private query is answered for a database that includes data about an individual I, then any bad (or good) outcome for I becomes at most e^{ε} more (or less) likely. ε is a tunable parameter that controls the strength of the privacy guarantee: smaller values of ε mean more privacy. [34] also offers an economic interpretation of ε values.

Of course, in practice we would like to ask more than one query. This is possible because differential privacy is *compositional*: answering two queries that are ε_1 and ε_2-differentially private, respectively, is no worse than answering a single $(\varepsilon_1 + \varepsilon_2)$-differentially private query [46]. In essence, we can think of the parameter ε as a *privacy budget*: we negotiate once with the users what setting of ε they feel comfortable with, and we can then answer an arbitrary set of ε_i-differentially private queries, as long as $\sum_i \varepsilon_i \leq \varepsilon$.

But what happens if the privacy budget is exhausted? In this case, PRISM would have to (forever) stop answering queries! However, we could avoid this undesirable outcome if a) the budget is very large, or b) there is a way to replenish the budget. We discuss each in turn.

4.2 How soon would the budget run out?

To estimate how many queries PRISM could answer, we use a simple model that was proposed in [52]. Suppose our privacy budget is ε and we would like to answer queries with sensitivity s using the Laplace mechanism, such that the noised answer is within $\pm E$ of the true answer with probability c. Then we can answer

$$N = \frac{\varepsilon \cdot E}{-2 \cdot s \cdot ln(1-c)}$$

queries. The value of ε depends on the users' preferences, but $\varepsilon = 1$ has been suggested in [21, 49, 47].

Size helps: The first factor that works in favor of PRISM is that the Internet is very big – and differential privacy works best for large amounts of data. For concreteness, let us assume that PRISM is typically asked counting queries ($s = 1$) about IP addresses, of which there are $4 \cdot 10^9$. If a typical true answer is around $4 \cdot 10^7$, and we would like the noisy answer to be within 10% of that with confidence $c = 95\%$, we can ask $N = 667,616$ queries – more than half a million! This is a lot, but, of course, there are also approximately 60,000 ASes, many of which would want to answer queries fairly regularly. Based on this calculation alone, each AS could only ask ten queries, which seems discouraging. But there are other factors that work in PRISM's favor.

Sampling helps: Another favorable fact is that, due to the enormous amount of data, it is often necessary to use sampling for scalability. For instance, the NetFlow functionality is often configured to sample flows or packets in a 1-in-N fashion. This helps with privacy, too: it is known [36] that, when sampling the data with a factor β (say, 1%), the privacy cost of the queries can also be scaled by β, since the data of each individual contributes to only one in $1/\beta$ samples on expectation. If we assume that NetFlow sampling uses a rate of $\beta = 1\%$, we can immediately scale the privacy budget by a factor of $1/\beta = 100$ and arrive at $1,000$ queries per AS. Sampling inevitably introduces imprecision, but estimating the statistics of a larger population with its subsamples is a well-studied topic.

If sampling is additionally applied to the NetFlow records themselves, we can further boost ε at the expense of some additional imprecision in the result (see also Section 5.2). For instance, the US census bureau uses a 1% Public Use Microdata Sample [1]. If we assume that $\beta' = 1\%$ is reasonable, we can boost the privacy budget by *another* factor of $1/\beta' = 100$. (Of course, it must not be revealed *which* individuals were sampled for which query; in practice, this could be implemented with a secure coin toss and a simple multiparty computation circuit [13].) With this, we arrive at $6.68 \cdot 10^9$ queries, or roughly $100,000$ per AS.

Competition between ISPs helps: Perhaps the biggest opportunity comes from the fact that differential privacy makes very pessimistic assumptions about collusion: once a query is answered, the recipient of the answer shares it with everyone. This assumption is prudent in some scenarios, but in the Internet, most ISPs are business competitors and have conflicting interests, so it seems unlikely that they would collude on a massive scale.

In principle, if responses received by one ISP could *never* make it to another ISP, we could give each ISP its own privacy budget of six billion queries. But mistakes happen, computers get hacked, and some ISPs might in-

deed collude on a small scale, so this is not entirely realistic. But even if each ISP eventually shares its responses with several other ISPs, it still seems possible to let each ISP answer several hundred million queries. Even at 1,000 queries per day, 400 million queries would last more than 1,000 years – far beyond the likely lifetime of PRISM, or even the Internet.

4.3 Can the budget be replenished?

So far, we have assumed that the budget is set once and for all, and can never be replenished. This is because differential privacy conservatively assumes that all the queries are answered based on the *same* data; it is designed to protect against a "worst case" in which the entire privacy budget is used to gain information about a single individual. However, in practice, much of the Internet's metadata is ephemeral: 40% of the /24-blocks are dynamically allocated, with a median re-allocation period of 2.5 hours [18], and most end-to-end Internet routes change within several hours [27]. Since the Internet's old metadata are being constantly replaced by new entries, we expect that its underlying databases will *eventually* be entirely new.

This high level of churn also applies at other levels. At the flow level, a GEANT router receives roughly 10^5 new flow records per second [26], and this is at a sampling rate of 1/1000. Previous studies also found that flows expire quickly ([14] reports that 45% flows expire within two seconds, and 98% within 15 minutes), and that longer flows are more likely to be computer-to-computer protocols that do not involve end users [54], which are presumably not as sensitive. At the user level, the IP-to-user mapping also changes over time, as users change ISPs or move to a different workplace.

Since PRISM's database is thus likely to be in constant flux, it does not seem unreasonable to replenish the privacy budget once in a while. Very conservatively, we could replenish the budget once every 100 years, since the database almost certainly contains different individuals by then; based on the above calculations, this would still be enough to answer 10,000 queries per ISP per day, and faster schedules are probably possible. For practical reasons, we might opt for a much smaller budget that is replenished much more frequently – say, a budget of 1,000 queries, with 100 added each day. This would limit the damage in case an ISP's PRISM account is compromised and the attacker chooses to burn the entire budget on a single query.

4.4 Summary

At first glance, it seems that a system like PRISM could only work for a short time, until its privacy budget is exhausted. However, due to the Internet's enormous size and the low likelihood of massive collusion between ISPs, the budget could last a very long time, possibly decades – and, due to the Internet's high rate of churn, it would probably be safe to replenish it periodically. Indeed, the numbers are large enough that, even if some of our arguments were dismissed, the rest would still be enough to show that PRISM is feasible.

5 Discussion

In this section, we discuss other questions about PRISM and a potential "knowledge plane".

5.1 Could PRISM protect ISP privacy?

So far, our discussion has focused on the privacy of *individuals*, which we have taken to mean individual users who connect to the Internet. However, it is clear that ISPs, too, are concerned about privacy. For instance, an ISP might be concerned that, if its topology or traffic matrix were revealed, other ISPs might use this information to gain a competitive advantage. This concern is frequently discussed in papers on active and passive measurement techniques [65, 59, 60], and there is evidence that at least some ISPs have taken steps to discourage probing, e.g., by ICMP rate limiting [65].

If PRISM does not address these concerns, ISPs may choose to share only data they would publish anyway, or they may even choose not to participate at all. However, there is a way to enforce ISP privacy in PRISM: it can consider other database schemata in which rows contain the data from entire subnets, PoPs, or even entire ISPs, and then add noise according to the schema that is the most restrictive.[1] It is true that some information could still be learned, e.g., a very rough traffic matrix – differential privacy is *meant* to release large trends like this! However, high-level information like that can often be inferred remotely anyway, as systems like Hubble [38], DisCarte [62], etc., have repeatedly shown. Indeed, these systems typically yield far more specific data than PRISM ever would.

So, on the one hand, PRISM would substantially broaden the range of queries that could be asked, and it might yield data at a higher quality than measurement systems would, since it operates on the true data, without heuristics-based inference or measurement errors. But, on the other hand, ISPs would gain more explicit control over the information they are sharing (including an option to audit queries that involve their local node, and to block queries that are abusing the system).

[1] Note that we do not actually need multiple databases – it is sufficient to consider the schemata when deciding how much noise to add to a given query result.

5.2 Would PRISM be accurate enough?

Differential privacy inherently returns imprecise results, so it is natural to ask how this imprecision would affect utility. However, dealing with imprecision is a challenge in almost all quantitative studies. For instance, there is often too much data to collect, so sampling has to be used, which introduces sampling error; indeed, sampling is a built-in part of NetFlow for that very reason [6]. Other sources of error include "gaps" in the data because of partial deployment or partial visibility, data quality issues, samples from slightly different points in time, etc. The effects of this imprecision are well understood, and there are excellent statistical tools and techniques that can be used to deal with them, which are standard practice for measurement studies. In a sense, PRISM's "noise" might even be cleaner than that in many measurements today: PRISM would operate on the true data, so there would be less inaccuracy due to heuristics-based inferences, data quality issues, or assumptions that do not hold everywhere. Moreover, PRISM's "noise" would be drawn from a well-known distribution, so it would be easier to reason about its impact on the data.

5.3 What about data quality issues?

Unlike classical measurement studies, PRISM would return only the final answer (e.g., the average traffic on a given set of links), but not return intermediate results (the traffic on the individual links). This would make it harder to spot problems with the data. For instance, if a router is misconfigured and reports Exabits of traffic per second, the overall result will be completely implausible, but there will not be an easy way to identify the problem.

One way to address this problem would be to encode the quality-checking in the query itself. In Section 3.1, we have used a simple SQL-like syntax, but there are far more sophisticated query languages for differentially private data analysis, including higher-order languages [57]. Recall that the query has access to the true, un-noised data while it is being processed, so it can perform plausibility checks and data cleaning internally; if a problem with a particular row of data is too severe, the query might return a default value for that row. To assess the quality of the data, a second query could be issued to count the number of problematic rows.

Since PRISM would be operated by ISPs, intentional tampering with the results does not seem very likely. Nevertheless, we note that there are ways to address this if it should become an issue, e.g., by (privately) enforcing upper and lower bounds on the data each ISP can return, analogous to PDDP [21]. Another possibility would be to enforce differential privacy in a verifiable manner, similar with VerDP [51].

5.4 How well does query processing scale?

In principle, PRISM could be used to run queries across the entire Internet; however, we expect that, in practice, many queries will involve only a small number of ISPs. For example, [22] says that results from four domains are sufficient to detect DDoS with 98% accuracy; another study [16] only used data from a network's 17 customers.

Whether a query can scale depends on the operators it contains. If a query can be broken into per-ISP subqueries, the subqueries can be processed in parallel, and the final aggregation step is fairly easy. If joins are involved, more expensive cryptographic techniques like private set intersection [31, 52] will probably need to be used, and these do not yet scale too well. However, this is an active and fairly young field of research, so PRISM could benefit from new discoveries over time. For example, SEPIA [45] was able to intersect sets from 25 players with one million elements in slightly more than 1 minute; this seems like a practical scale for at least some queries.

5.5 Would a partial deployment work?

Like many other systems, PRISM would initially face a chicken-and-egg problem because its utility depends in part on the amount of data that is available for querying. Nevertheless, the situation is not as dire as, e.g., in S-BGP: even a small deployment of two or three large ISPs could initially be useful, e.g., to privately find ways to optimize traffic between them, or to privately track down attacks. Other ISPs would have an incentive to join because this would give them the ability to ask queries of their own.

Despite all this, it would be naïve to expect a full deployment, or that all PRISM nodes are always available. This should not be a problem for most queries, however: if a query is limited to a moderate set of ASes, it can be answered by the PRISM nodes in these ASes, without involving the others; sampling-based queries could be run using samples from other ISPs that are currently available. Global queries, or queries that require a truly unbiased sample, could be processed in stages.

Some queries might return a biased answer if they are answered based on data from a subset of ISPs. However, working with partial data is a familiar problem for many administrators (e.g., from working with Internet looking glasses) and researchers doing measurement studies. Moreover, the set of ISPs that operate PRISM nodes could probably be made public, just as the set of RouteViews nodes and looking glasses is public; this should help with query design.

6 Related Work

Analyzing the Internet's sensitive metadata with privacy guarantees has been used for distributed troubleshooting [17, 16], measurement [58, 60, 59, 28], forecasting [9], route computation [32], and private alerts correlation [70]. There has also been initial attempts to apply differential privacy [29] – the strongest privacy model – to centralized [51, 47, 46] and distributed [52, 21] data sources. Our work puts differential privacy in the context of forming an Internet knowledge plane, and explores a common concern whether differential privacy's limited budget restricts its practicality [33, 56, 21, 20].

Creating a "knowledge plane" for the Internet is a longstanding proposal [24]. Many existing papers advance this goal by, e.g., measuring [43, 44] or predicting [44] Internet performance, developing an Internet-scale query processor [35], a scalable distributed information management system [69], a framework for Internet forensics [61], or a declarative programming environment [67]. In contrast, our work focuses less on a concrete design and more on providing strong privacy guarantees, which would be important for any practical knowledge plane.

7 Conclusions

We have made a case for differential privacy to be a potential basis for realizing the long-standing vision of a "knowledge plane" for the Internet. We have sketched the design of a system called PRISM that could serve as the foundation of such a knowledge plane; and we have especially focused on a common concern of differential privacy's limited budget. PRISM may not be able to support *all* queries we might like to ask, but it would certainly be able to answer a wide range of queries – completely automatically and, unlike its namesake, with one of the strongest privacy guarantees available today. Although we have mainly used NetFlow records as examples, the idea of PRISM should be generalizable to other types of network data, too.

References

[1] 1-Percent Public Use Microdata Sample. http://www.census.gov/census2000/PUMS.html.

[2] Arbor Networks. http://www.arbornetworks.com/netflow-analysis.html.

[3] NetFlow Auditor. http://www.netflowauditor.com/.

[4] NetFlow support. https://www.manageengine.com/products/netflow/help/cisco-netflow/netflow-ios-versions.html.

[5] PRISM. http://www.theverge.com/2013/7/17/4517480/nsa-spying-prism-surveillance-cheat-sheet.

[6] Sampled NetFlow. http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/12s_sanf.pdf.

[7] M. Allman, E. Blanton, V. Paxson, and S. Shenker. Fighting coordinated attackers with cross-organizational information sharing. In *Proc. HotNets*, 2006.

[8] D. Antoniades, S. Antonatos, and E. Markatos. Flexible and high-performance anonymization of NetFlow records using Anontool. In *Proc. SecureComm*, 2007.

[9] M. Atallah, M. Bykova, J. Li, K. Frikken, and M. Topkara. Private collaborative forecasting and benchmarking. In *Proc. WPES*, 2004.

[10] A. J. Aviv and A. Haeberlen. Challenges in experimenting with botnet detection systems. In *Proc. CSET*, 2011.

[11] M. Barbaro and T. Zeller. A face is exposed for AOL searcher no. 4417749. http://nytimes.com/2006/08/09/technology/09aol.html.

[12] R. M. Bell and Y. Koren. Lessons from the Netflix prize challenge. *SIGKDD Explor. Newsl.*, 9(2):75–79, Dec. 2007.

[13] A. Ben-David, N. Nisan, and B. Pinkas. FairplayMP - a system for secure multi-party computation. In *Proc. CCS*, 2008.

[14] N. Brownlee and kc claffy. Understanding Internet traffic streams: Dragonflies and tortoises. *IEEE Communications Magazine*, 40(10):110–117, Oct. 2002.

[15] N. Brownlee and D. Plonka. IP flow information export (IPFIX). http://datatracker.ietf.org/wg/ipfix/charter/.

[16] M. Burkhart and X. Dimitropoulos. Privacy-preserving distributed network troubleshooting – bridging the gap between theory and practice. In *TISSEC*, 2011.

[17] M. Burkhart, M. Strasser, D. Many, and X. Dimitropoulos. SEPIA: Privacy-preserving aggregation of multi-domain network events and statistics. In *Proc. USENIX Security*, 2010.

[18] X. Cai and J. Heidemann. Understanding block-level address usage in the visible Internet. In *Proc. SIGCOMM*, 2010.

[19] S. Chaudhuri, V. Ganti, and R. Kaushik. A primitive operator for similarity joins in data cleaning. In *Proc. ICDE*, 2006.

[20] R. Chen, G. Acs, and C. Castelluccia. Differentially private sequential data publication via varible-length n-grams. In *Proc. CCS*, 2012.

[21] R. Chen, A. Reznichenko, P. Francis, and J. Gehrke. Towards statistical queries over distributed private user data. In *Proc. NSDI*, 2012.

[22] Y. Chen, K. Hwang, and W.-S. Ku. Collaborative detection of DDoS attacks over multiple network domains. *TPDS*, 18(12):1649–1662, Dec. 2007.

[23] E. B. Claise. Cisco systems NetFlow services export version 9. RFC 3954 (Proposed Standard), Oct. 2004.

[24] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski. A knowledge plane for the Internet. In *Proc. SIGCOMM*, 2003.

[25] A. Cooper. Report from the Internet privacy workshop. RFC 5389 (Proposed Standard), Jan. 2012.

[26] I. Cunha, F. Silveira, R. Oliveira, R. Teixeira, and C. Diot. Uncovering artifacts of flow measurement tools. In *Proc. PAM*, 2009.

[27] I. Cunha, R. Teixeira, and C. Diot. Measuring and characterizing end-to-end route dynamics in the presence of load balancing. In *Proc. PAM*, 2011.

[28] M. Djatmiko, D. Schatzmann, A. Friedman, X. Dimitropoulos, and R. Boreli. Privacy preserving distributed network outage monitoring. In *Proc. INFOCOM Poster*, 2013.

[29] C. Dwork. Differential privacy. In *Proc. ICALP*, 2006.

[30] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proc. TCC*, 2006.

[31] M. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *Proc. EUROCRYPT*, May 2004.

[32] D. Gupta, A. Segal, A. Panda, G. Segev, M. Schapira, J. Feigenbaum, J. Rexford, and S. Shenker. A new approach to interdomain routing based on secure multi-party computation. In *Proc. HotNets*, 2012.

[33] A. Haeberlen, B. C. Pierce, and A. Narayan. Differential privacy under fire. In *Proc. USENIX Security*, Aug. 2011.

[34] J. Hsu, M. Gaboardi, A. Haeberlen, S. Khanna, A. Narayan, B. C. Pierce, and A. Roth. Differential privacy: An economic method for choosing epsilon. In *Proceedings of the 2014 IEEE Conputer Security Foundations Symposium (CSF'14)*, July 2014.

[35] R. Huebsch, J. M. Hellerstein, N. Lanham, B. T. Loo, S. Shenker, and I. Stoica. Querying the Internet with PIER. In *Proc. VLDB*, 2003.

[36] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? In *Proc. FOCS*, 2008.

[37] E. Katz-Bassett, H. Madhyastha, V. Adhikari, C. Scott, J. Sherry, P. van Wesep, A. Krishnamurthy, and T. Anderson. Reverse traceroute. In *Proc. NSDI*, 2010.

[38] E. Katz-Bassett, H. V. Madhyastha, J. P. John, A. Krishnamurthy, D. Wetherall, and T. Anderson. Studying black holes in the Internet with Hubble. In *Proc. NSDI*, 2008.

[39] E. Kenneally and k. claffy. Dialing privacy and utility: a proposed data-sharing framework to advance Internet research. In *Proc. IEEE Security and Privacy*, 2010.

[40] A. Lakhina, M. Crovella, and C. Diot. Characterization of network-wide anomalies in trafc flows. In *Proc. IMC*, 2004.

[41] M. Lee, N. Duffield, and R. R. Kompella. Two samples are enough: Opportunistic flow-level latency estimation using NetFlow. In *Proc. INFOCOM*, 2010.

[42] A. Li, X. Liu, and X. Yang. Bootstrapping accountability in the Internet we have. In *Proc. NSDI*, 2011.

[43] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iPlane: An information plane for distributed services. In *Proc. OSDI*, 2006.

[44] H. V. Madhyastha, E. Katz-Bassett, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iPlane Nano: Path prediction for peer-to-peer applications. In *Proc. NSDI*, 2009.

[45] D. Many, M. Burkhart, and X. Dimitropoulos. Fast private set operations with SEPIA. Technical Report TIK-Report No. 345, Communication Systems Group, ETH Zurich, Switzerland, 2012.

[46] F. McSherry. Privacy integrated queries. In *Proc. SIGMOD*, June 2009.

[47] F. McSherry and R. Mahajan. Differentially-private network trace analysis. In *Proc. SIGCOMM*, 2010.

[48] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *Proc. FOCS*, Oct. 2007.

[49] P. Mohan, A. Thakurta, E. Shi, D. Song, and D. E. Culler. GUPT: Privacy preserving data analysis made easy. In *Proc. SIGMOD*, 2012.

[50] D. Moore, G. M. Voelker, and S. Savage. Inferring Internet denial-of-service activity. In *Proc. USENIX Security*, 2001.

[51] A. Narayan, A. Feldman, A. Papadimitriou, and A. Haeberlen. Verifiable differential privacy. In *Proceedings of EuroSys 2015*, Apr. 2015.

[52] A. Narayan and A. Haeberlen. DJoin: Differentially private join queries over distributed databases. In *Proc. OSDI*, 2012.

[53] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, and H. Zhang. An empirical evaluation of entropy-based traffic anomaly detection. In *Proc. IMC*, 2008.

[54] L. Quan and J. Heidemann. On the characteristics and reasons of long-lived Internet flows. In *Proc. IMC*, 2010.

[55] V. Rastogi and S. Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proc. SIGMOD*, June 2010.

[56] J. Reed, A. J. Aviv, D. Wagner, A. Haeberlen, B. C. Pierce, and J. M. Smith. Differential privacy for collaborative security. In *Proc. EuroSec*, 2010.

[57] J. Reed and B. C. Pierce. Distance makes the types grow stronger: A calculus for differential privacy. In *Proc. ICFP*, 2010.

[58] F. Ricciato and M. Burkhart. Reduce to the max: A simple approach for massive-scale privacy-preserving collaborative network measurements. In *Proc. TMA*, 2011.

[59] M. Roughan and Y. Zhang. Privacy-preserving performance measurements. In *Proc. MineNet*, 2006.

[60] M. Roughan and Y. Zhang. Secure distributed data-mining and its application to large-scale network measurements. In *Proc. SIGCOMM CCR*, 2006.

[61] V. Sekar, Y. Xie, D. A. Maltz, M. K. Reiter, and H. Zhang. Toward a framework for Internet forensic analysis. In *Proc. HotNets*, 2004.

[62] R. Sherwood, A. Bender, and N. Spring. DisCarte: A disjunctive Internet cartographer. In *Proc. SIGCOMM*, 2008.

[63] E. Shi, T.-H. H. Chan, E. G. Rieffel, R. Chow, and D. Song. Privacy-preserving aggregation of time-series data. In *Proc. NDSS*, 2011.

[64] C. Soghoian. Surveillance and security lessons from the Petraeus scandal. ACLU blog, Nov. 2012.

[65] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with Rocketfuel. In *Proc. SIGCOMM*, 2002.

[66] M. B. Tariq, M. Motiwala, N. Feamster, and M. Ammar. Detecting general network neutrality violations with causal inference. In *Proc. CoNEXT*, 2009.

[67] M. Wawrzoniak, L. Peterson, and T. Roscoe. Sophia: An information plane for networked systems. In *Proc. HotNets*, 2003.

[68] E. Wustrow, M. Karir, M. Bailey, F. Jahanian, and G. Huston. Internet background radiation revisited. In *Proc. IMC*, 2010.

[69] P. Yalagandula and M. Dahlin. A scalable distributed information management system. In *Proc. SIGCOMM*, 2004.

[70] V. Yegneswaran, P. Barford, and S. Jha. Global intrusion detection in the DOMINO overlay system. In *Proc. NDSS*, 2004.

[71] Y. Zhang and V. Paxson. Detecting stepping stones. In *Proc. USENIX Security*, 2000.