Privacy-Friendly Collaboration for Cyber Threat Mitigation

Julien Freudiger†, Emiliano De Cristofaro‡, and Alex Brito†
†PARC ‡University College London

Abstract

In recent years, security solutions have increasingly focused on actively predicting future attacks. Since prediction accuracy can improve with more information about the attackers, sharing security-relevant data among organizations is often being advocated. However, collaborative security approaches are rarely implemented due to related trust, privacy, and liability concerns.

In this paper, we explore a novel approach to collaborative threat mitigation where organizations estimate the benefits of data sharing with potential partners in a privacy-preserving way (i.e., without actually disclosing their dataset). Data sharing then occurs securely within coalitions of allied organizations. We focus on collaborative predictive blacklisting, i.e., predicting sources of future attacks based on both one’s own data and that of a few selected partners. We study how different collaboration strategies affect prediction accuracy by experimenting on a real-world dataset of 2 billion IP addresses and observe up to a 105% improvement.

1. INTRODUCTION

Modern security solutions aim beyond passive defense and attempt to actively predict attacks by learning from offensive practices. Previous research has shown that their accuracy improves if organizations share data with each other, as attackers tend to target victims in similar ways [27, 44, 52]. Thus, collaboration among victims is often being advocated. Despite clear advantages, organizations are reluctant to share security-relevant data due to related trust, privacy, and liability concerns. Corporate data such as firewall logs might expose confidential information, challenge a corporation’s competitiveness, or reveal negligence.

To address these issues, previous work proposed to sanitize data prior to sharing [1, 31, 37, 43, 49], e.g., via generalization and suppression. However, this makes data less useful [30], and still prone to de-anonymization [8]. Other work suggested reporting encrypted data to a semi-trusted central repository that obliviously aggregates contributions [3], or relying on distributed data aggregation protocols based on secure multi-party computation and secret sharing [6]. While aggregation can help detect malicious activities or compute traffic statistics, it operates akin to large alert repositories as it identifies most prolific attack sources and yields global models. As shown in [44, 52], however, generic attack models are not desirable as they miss a significant number of attacks, especially when attack sources choose targets strategically and focus on a few known vulnerable networks.

In theory, organizations could provide encrypted data to a central authority that obliviously computes personalized recommendations for threat mitigation using Fully Homomorphic Encryption (FHE) [18]. However, FHE is still far from being practical and it remains unclear whether complex machine learning algorithms needed for the prediction could run effectively over encrypted data and reveal personalized results to data owners only.

Intuition. We propose an alternative approach to collaborative threat mitigation where organizations are able to find suitable collaboration partners in a distributed and privacy-preserving way, and organize into coalitions prior to data sharing. This way, data sharing takes place within groups of related victims that partner up with relevant sources of information and obtain data that improves their predictions. In our envisioned model, parties first identify a set of potential partners from a larger pool of organizations, e.g., corporations in the same sector, and then select the best partners within the corresponding set. In practice, this can be repeated over time to ensure relevant and near real-time protection.

To demonstrate the feasibility of this approach, we experiment with a framework called Sharing is Caring (SIC), which supports two types of algorithms: one for estimating the benefits of sharing data with a potential partner in a privacy-preserving way (i.e., without disclosing plaintext data), and the other for sharing agreed-upon datasets with selected partners, e.g., only common attacks.

We focus on collaborative predictive blacklisting, namely predicting attack sources based on logs generated by firewalls and/or intrusion detection systems (see Sec. 2.3). Collaboration can improve accuracy since organizations rely not only on their own logs, but also on data from selected partners, which SIC helps identify in a privacy-friendly way. Privacy is provably guaranteed, as benefit estimation and information sharing occur via secure computation.

Experiments. One of our main goals is to investigate which collaboration strategies work best, in terms of the resulting improvement in prediction accuracy. To this end, we conduct several experiments on a real-world dataset of 2 billion suspicious IP addresses collected by DShield.org [42] over 2 months (see Sec. 4). This dataset contains a large variety of contributors, as confirmed in our analysis, which allows us to test the effectiveness of data sharing among diverse groups of victims.

We perform a quantitative analysis on this dataset in order to identify victims’ and attackers’ profiles, among other features. This helps us clean the dataset and design a meaningful data sharing experiment. We repeatedly select 100 victims at random from regular contributors and measure the efficacy of our techniques in a scenario where collaboration occurs within small coalitions. The experiment setting models a scenario where a highly diverse community of contributors attempts to form coalitions. We then test multiple collaboration strategies using a standard prediction algorithm based on Exponentially Weighted Moving Average (EWMA) [44].

Main Results. Our analysis yields several key findings in the scope of private collaboration for threat mitigation. Specifically, we observe that: (1) the more information is available about attackers, the
better the prediction, as intuitively expected; (2) different collaboration strategies yield a large spectrum of performances, in fact, with some strategies, sharing does not actually help much; (3) sharing information only about common attackers is almost as useful as sharing everything. This highlights both the importance of selecting the right partners and the usefulness of limited data sharing.

Contributions. To the best of our knowledge, our work is the first to provide a privacy-friendly solution for collaborative predictive blacklisting. We demonstrate that data sharing does not have to be an “all-or-nothing” process; by relying on efficient cryptographic protocols for privacy-preserving information sharing, it is possible to only share relevant data, and only when beneficial. Compared to prior work, our approach has several important advantages: (1) it helps (privately) identify entities with good partnership potential, (2) it minimizes information disclosure, and (3) it increases speed of malicious activity detection, leading to near-real-time mitigation.

Our work could also be applied to other security-related applications that benefit from data sharing, such as spam filtering [11], virus detection [21], or DDoS mitigation [36].

2. PRELIMINARIES

2.1 System Model

We assume a network of entities $V = \{V_i\}_{i=1}^n$, with each $V_i$ maintaining a dataset $S_i$ of suspicious events, such as suspicious IP addresses observed by a firewall (IP, time, port). We denote this list of events as $L_i$ (for each entity $V_i$). Hence, $S_i = \{L_i\}$. Each entity $V_i$ aims to predict and block (i.e., blacklist) future attacks.

Existing Approaches. Thus far, two main approaches have been used for predictive blacklisting: (1) no collaboration, i.e., each entity $V_i$ independently performs the prediction based only on its own dataset $S_i$, or (2) community-based, i.e., each entity $V_i \in V$ submits its dataset $S_i$ to a central repository, which returns a customized blacklist for $V_i$, also based on every entity’s dataset. The latter provides increased accuracy [52, 44] but requires entities to reveal their entire datasets to a central repository.

Our Novel Model. We introduce a privacy-friendly collaborative model for predictive blacklisting, whereby entities identify good collaboration partners via pairwise secure computations (without the need for a trusted third-party), and then share data. This way, data sharing takes place in groups of related victims. Each entity performs predictions based not only on its own dataset but also on an augmented dataset that comprises information possibly shared by the counterpart. This peer-to-peer model seeks to combine the best of the two aforementioned models: improved prediction and no wholesale disclosure of datasets. To this end, we rely on efficient cryptographic protocols for privacy-preserving information sharing, presented below.

Threat Model. We denote with $A \in V$ an adversary attempting to learn information about other entities’ datasets. (External adversaries are not considered, since their actions can be mitigated via standard network security techniques.) In the worst case, $A$ may try to collaborate with all other entities and collect available information after each data sharing attempt. $A$ obtains network traces that allow inference of strategic information. Hence, we aim to protect data confidentiality for each $V_i \in V$. We assume adversary $A$ to be semi-honest (or honest-but-curious). $A$ follows protocols’ specifications and does not misrepresent any of its inputs. However, during or after protocol execution, $A$ might attempt to infer additional information about other parties’ inputs.

2.2 Privacy-preserving Information Sharing

We now review a few cryptographic primitives used throughout the rest of the paper.

Secure Two-Party Computation (2PC) allows two parties, on input $x$ and $y$, respectively, to privately compute the output of a public function $f$ over $(x, y)$. Both parties learn nothing beyond what can be inferred from the output of the function. For more details on 2PC, and implementations thereof, we refer to [50, 19, 24].

Private Set Intersection (PSI) involves two parties, a server, on input a set $S$, and a client, on input a set $C$. At the end of the interaction, the latter only learns $S \cap C$, whereas the former learns nothing beyond client’s set size. State-of-the-art instantiations, with different complexities and computational assumptions, include both garbled-circuit based techniques [23, 16] and specialized protocols [17, 29, 14, 26, 15]. In our experiments, we use the PSI construction presented in [14], secure under the One-More-RSA assumption [4] in the Random Oracle Model (ROM), with computational and communication complexities linear in set sizes. Note, however, that one can select any PSI construction, without affecting our design.

Private Set Intersection Cardinality (PSI-CA) involves two parties, a server, on input a set $S$, and a client, on input a set $C$. At the end of the interaction, the latter only learns $|S \cap C|$, and the server learns nothing beyond client’s set size. PSI-CA is a more “stringent” variant than PSI, as it only reveals the magnitude of the intersection, but not the actual contents. There are several instantiations of PSI-CA [17, 2, 22, 13], and, in our experiments, we use the construction presented in [13], which has linear complexities, with security under the One-More-DH assumption [4] in the Random Oracle Model (ROM). Again, note that any PSI-CA construction can be employed.

Private Jaccard Similarity (PJS) involves two parties, a server, on input a set $S$, and a client, on input a set $C$. At the end of the interaction, the client only learns $J(S, C) = \frac{|S \cap C|}{|S| + |C| − |S \cap C|}$, where $J(S, C)$ denotes the Jaccard Similarity index [25] between sets $S$ and $C$. Blundo et al. [5] slightly relax the above definition and shows how to privately compute the Jaccard Similarity index using only PSI-CA. Since $J(S, C) = \frac{|S \cap C|}{|S| + |C| − |S \cap C|}$, parties can obtain $J(S, C)$ without disclosing actual set contents, but only the intersection cardinality, which, however, reveals slightly more information than only the intersection/union ratio.

2.3 Predictive Blacklisting

Prediction algorithms rely on past data to decide on likely future events. In this paper, we focus on collaborative predictive blacklisting [52]. The goal is to forecast future potential malicious sources based on past attacks. For instance, each organization predicts the likelihood of future attacks from particular IP addresses.

Algorithm. Let $t$ denote the day an attack was reported and $T$ the current time, so $t = 1, 2, ..., T$. We partition $T$ into two windows of consecutive days: a training window $T_{\text{train}}$ and a testing window, $T_{\text{test}}$. Prediction algorithms rely on information in the training data, $t \in T_{\text{train}}$, to tune their model and validate the predictions for the testing data, $t \in T_{\text{test}}$.

The Global Worst Offender List (GWOL) is a basic prediction algorithm that selects top attack sources from $T_{\text{train}}$, i.e., highest number of globally reported attacks [52]. Local Worst Offender List (LWOL) is the local version of GWOL and operates on a local network based entirely on its own history [52]. These approaches encounter some limitations as LWOL fails to predict on attackers...
not previously seen, while GWOL tends to be irrelevant to small victims. Thus, machine learning algorithms were suggested to improve GWOL and LWOL [44, 52].

We use the Exponentially Weighted Moving Average (EWMA) algorithm, as proposed by Soldo et al. [44] to perform blacklisting prediction. EWMA uses time series aggregation: it consists in aggregating attack events from \( T_{\text{train}} \) to predict future attacks. Other features one could consider include the historical malicious activity of an IP address, the clustering of IP addresses with similar malicious behavior, and the network centrality of a target. Observe that it is out of the scope of this paper to improve on existing prediction algorithms. Instead, we focus on how to help organizations identify useful partners in a privacy-preserving way, and how different collaboration strategies perform in comparison to each other.

Accuracy Metrics. As commonly done with prediction algorithms, we measure accuracy with True Positives (TP), which is the number of predictions that correctly match future events. In the blacklisting scenario, TP correspond to the number of attacks in the blacklist that are correctly predicted.

In practice, sources might not be blacklisted at once and blacklisting algorithms might rely on several observations over time before blacklisting a source, such as the rate at which the source is attacking, the payload of suspicious packets, etc. It is important to distinguish between the prediction algorithm, which identifies potential malicious sources and/or creates a watch-list from the blacklisting algorithm, which actually blocks sources. Blacklisting algorithms are site-specific and need to optimize, among others, false negative and false positive ratios. The prediction algorithm enables the identification of suspicious IP addresses that deserve further scrutiny and improve the effectiveness of blacklisting algorithms.

Therefore, just like prior work [44, 52], we focus on measuring the TP of the prediction algorithm, i.e., the ability to identify potential sources of attacks, and do not consider false positives as it is out of the scope of our work.

Upper Bounds. A future attack can be predicted if it already appeared in the logs of some victims. Traditional upper-bounds on collaboration algorithms capture this and we use them to evaluate the performance of our collaboration algorithms. The Global Upper Bound (GUB) measures, for every target \( V_i \), the number of attackers that are both in the training window of any victim and in \( V_i \)’s testing window. For every \( V_i \), we define the Local Upper Bound (LUB) \( LUB(V_i) \), as the number of attackers that are both in \( V_i \)’s training and testing windows. Note that one could also predict attackers that have never been reported before (e.g., when an entire IP subnet is compromised), but this is out of the scope of this paper.

3. THE SIC FRAMEWORK

3.1 Overview

We describe the logic of the Sharing Is Caring (SIC) framework in two steps, presenting algorithms supporting the secure selection of collaboration partners, and algorithms for the privacy-preserving merging of (i.e., sharing) datasets among partners. As discussed in Sec. 2, we assume a network of \( |V| \) entities. We define \( S_i \) to be the set of unique IP addresses held by \( V_i \): \( S_i = \{ \text{IP} \in L_i \} \).

A high-level sketch of the SIC framework is presented in Fig. 1. In (1), potential partner entities \( V_i \) and \( V_j \) estimate the benefits they would receive from sharing their security data with each other. They could do so by securely computing one or multiple metrics. In (2), based on the estimated benefits, entities decide whether to partner or not. For instance, \( V_i \) and \( V_j \) become partners if the expected benefit is above a certain threshold; alternatively, each entity might partner with \( k \) other entities that yield the maximum benefits. Finally, in (3), partners merge their datasets, e.g., by only sharing common attacks and nothing else.

3.2 Select

Entities select collaboration partners by evaluating, in pairwise interactions, the potential benefits of sharing their data with each other. That is, potential benefits decide partnerships. This is done in a privacy-preserving way, as only a measure of anticipated benefits is revealed and nothing about datasets’ content.

Supported Metrics. We consider several similarity metrics for partner selection. Metrics are reported in Table 1, along with the corresponding protocols for their privacy-preserving computation. We consider similarity metrics as previous work [27, 52] showed that collaborating with correlated victims works well. Victims are correlated if they are targeted by correlated attacks, i.e., attacks mounted by the same source IP against different networks around the same time. Intuitively, correlation arises from attack trends; in particular, correlated victim sites may be on a single hit list or might be natural targets of a particular exploit (e.g., PHP vulnerability). Then, collaboration helps re-enforce knowledge about an on-going attack and/or learn about an attack before it hits.

Set-based and Correlation-based Similarity. We consider two set-based metrics: Intersection-Size and Jaccard, which measure set similarity and operate on unordered sets. We also consider Pearson and Cosine, which provide a more refined measure of similarity than set-based metrics, as they also capture statistical relationships. The last two metrics operate on data structures representing attack events, such as a binary vector, e.g., \( S_i = [s_{i1} \ s_{i2} \ \ldots \ s_{in}] \), of all possible IP addresses with 1-s if an IP attacked at least once and 0-s otherwise. This can make it difficult to compute correlation in practice, as both parties need to agree on the range of IP addresses under consideration to construct vector \( S_i \). Considering the entire range of IP addresses is not reasonable (i.e., this would require a vector of size 3.7 billion, one entry for each routable IP address). Instead, parties could either agree on a range via 2PC or fetch pre-defined ranges from a public repository.

In practice, entities could decide to compute any combination of metrics. In fact, the choice of metrics could periodically be re-
negotiated. Also, the list of metrics reported in Table 1 is non-exhaustive and others could be considered, e.g., for problems and datasets of different nature, as long as there exist a practical technique to securely evaluate them. Moreover, the benefits of collaboration might depend on other factors such as the amount and type of data merged, as well as the reputation of other parties.

Establishing Partnerships. After assessing the potential benefits of data sharing, entities make an informed decision as to whether or not to collaborate. Possible decision strategies include:

1. Threshold-based: \( V_i \) and \( V_j \) partner up if the estimated benefit of sharing is above a certain threshold;
2. Maximization: \( V_i \) and \( V_j \) independently enlist \( k \) potential partners to maximize their overall benefits (i.e., \( k \) entities with maximum expected benefits);
3. Hybrid: \( V_i \) and \( V_j \) enlist \( k \) potential partners to maximize their overall benefits, but also partner with entities for which estimated benefits are above a certain threshold.

In practice, entities might refuse to collaborate with other entities that do not generate enough benefits. One solution is to rely on well-known collaboration algorithms that offer stability (e.g., Stable Marriage/Roommate Matching [20]). Without loss of generality, we leave this for future work and assume cooperative parties: entities systematically accept collaboration requests.

Symmetry of Benefits. Some of the protocols used for secure computation of benefits, such as PSI-CA [13] and PJS [5], reveal the output of the protocol to only one party. Without loss of generality, we assume that this party always reports the output to its counterpart. We operate in the semi-honest model, thus parties are assumed not to prematurely abort protocols. Metrics discussed above are symmetric, i.e., both parties obtain the same value, and facilitate partner selection as both parties have incentive to select each other.

### 3.3 Merge

After the Select stage, entities are organized into coalitions, i.e., groups of victims that agreed to share data with each other. Entities can now merge their datasets with selected partners.

**Strategies.** Partners could share their datasets in several ways: e.g., they can disclose their whole data or only share which IP addresses they have in common, or transfer all attack events associated to common addresses and/or a selection thereof.

**Privacy-preserving Merging.** Our goal is to ensure that nothing about datasets is disclosed to partners beyond what is agreed. For instance, if partners agree to only share information about attackers they have in common, they should not learn any other information. Possible merging strategies, along with the corresponding privacy-preserving protocols, are reported in Table 2. Again, we assume that the output of the merging protocol is revealed to both parties.

Strategies denoted as Intersection/Union with Associated Data mean that parties not only compute and share the intersection (resp., union), but also all events related to items in the resulting set. Obviously, Union with Associated Data does not yield any privacy, as all events are mutually shared.

Note that organizations could limit the information sharing in time, e.g., by only disclosing data older than a month or of the last week. Also, previously proposed sanitization techniques [1, 31, 37, 43] could be used on top of SIC’s merging strategies.

### 3.4 Properties

**Privacy.** Our approach guarantees privacy through limited information sharing. Only data explicitly authorized by parties, and of interest to other parties, is actually shared. Therefore, the threat of information leakage is reduced. Data sharing occurs by means of secure two-party computation techniques, thus, security follows, provably, from that of underlying cryptographic primitives.

**Authenticity.** Recall that we assume semi-honest adversaries, i.e., entities do not alter their input datasets. If one relaxes this assumption, then it would become possible for a malicious entity to inject fake inputs or manipulate datasets to violate counterpart’s privacy. Nonetheless, we argue that assuming honest-but-curious entities is realistic in our model. First, organizations can establish long-lasting relations and reduce the risk of malicious inputs as misbehaving entities will eventually get caught. Second, given SIC’s peer-to-peer nature, one could also leverage peer-to-peer techniques to detect malicious behavior [40].

**Incentives and Competitiveness.** Since data exchanges are bi-directional, each party directly benefits from participation and can quantify the contribution of its partners. If collaboration metrics do not indicate high potential, each entity can deny collaboration. In other words, the incentive to participate is immediate as benefits can be quantified before establishing partnerships.

**Trust.** SIC relies on data to establish trust automatically, as previously explored by the peer-to-peer community [40]. If multiple entities report similar data, then it is likely correct and contributors can be considered as trustworthy. SIC enables entities to estimate each others’ datasets and potential collaboration value. This added transparency increases awareness of the contribution value and enables automation of trust establishment.

**Speed.** Due to the lack of a central authority (or vetting processes), collaboration and data sharing in SIC are instantaneous. Thus, it is possible for entities to interact as often and as fast as they wish.

### Table 1: Metrics for estimating potential benefits of data sharing between \( V_i \) and \( V_j \), along with corresponding protocols for their secure computation. \( \mu_i, \mu_j \) and \( \sigma_i, \sigma_j \) denote, resp., mean and standard deviation of \( S_i \) and \( S_j \).

| Benefit Estimation Metric | Operation | Private Protocol |
|---------------------------|-----------|------------------|
| Intersection Size | \(|S_i \cap S_j|\) | PSI-CA [13] |
| Jaccard | \(|S_i \cap S_j|/|S_i \cup S_j|\) | PJS [5] |
| Pearson | \[\sum_{i=1}^{N}(s_i - \mu_i)(s_j - \mu_j)/\sigma_i\sigma_j\] | Garbled Circuits [24] |
| Cosine | \[S_i S_j/|S_i||S_j|\] | Garbled Circuits [24] |

### Table 2: Strategies for merging datasets among partners \( V_i \) and \( V_j \), along with corresponding protocols for their secure computation.

| Sharing Strategy | Operation | Private Protocol |
|------------------|-----------|------------------|
| Intersection | \(S_i \cap S_j\) | PSI-CA [13] |
| Intersection with Associated Data | \{IP, (time, port)\} \(IP \in S_i \cap S_j\) | PSI with Data Transfer [14] |
| Union with Associated Data | \{IP, (time, port)\} \(IP \in S_i \cup S_j\) | – No Privacy – |
4. THE DSHIELD DATASET: OVERVIEW AND KEY CHARACTERISTICS

In order to assess the effectiveness of our approach, we should ideally obtain security data from real-world organizations. Such datasets are hard to come by because of their sensitivity. Therefore, we turn to DShield.org [42] and obtain a dataset of firewall and IDS logs mostly contributed by individuals and small organizations. DShield contains data contributors are willing to report, however, as in previous work [44, 52], we can assume strong correlation between the amount of reporting and the amount of attacks.

In this section, we show that DShield dataset contains data from a large variety of contributors (in terms of amount of contributions) and provides a reasonable alternative to experiment with our privacy-friendly collaborative approach.

4.1 The Dataset

We obtained two months’ worth of logs from the DShield repository. Each entry in the logs includes a Contributor ID, a source IP address, a target port number, and a timestamp (see Table 3).

| Contributor ID | Source IP      | Target port | Timestamp       |
|----------------|----------------|-------------|-----------------|
| 44cc551a       | 211.144.119.042| 1433        | 2013-01-01 11:48:36|

Table 3: Example of an entry in the DShield dataset.

The source of an attack refers to the attacker and target (or contributor) refers to a victim (V_i). Note that DShield anonymized the “Contributor ID” field by replacing it with a random yet unique string that maps to a single victim. Data obtained from DShield consists of about 2 billion entries, from 800K unique contributors, including more than 16M malicious IP sources, for a total of 170GB. We pre-processed the dataset in order to reduce noise and erroneous entries, following the same methodology adopted by previous work on DShield data [44, 52]. We removed approximately 1% of all entries, which belonged to invalid, non-routable, or unassigned IP addresses, or referred to non-existent port numbers (e.g., > 65536).

4.2 Measurements & Observations

We present a measurement analysis of the DShield dataset, aiming to better understand characteristics of attackers and victims. Overall, our observations are in line with prior work [10, 38, 39, 44] and underline the trend towards attackers hitting victims in a coordinated fashion, thus confirming the potential of collaboration among victims.

General Statistics. Fig. 2(a) shows the CDF of the fraction of victims that contribute their logs to DShield over the course of two months. We observe that about 75% of targets contribute less than 10% of the time, while about 6% of targets (50,000 targets) contribute daily. We describe at the end of this section how we filter out targets that seldom contribute.

Fig. 2(a) presents the histogram of the number of attacks per day, indicating about 30M daily attacks. We observe a significant increase around day 50 to 100M attacks. Careful analysis reveals that a series of IP addresses start to attack more aggressively around day 50, indicating what might be the beginning of a DoS attack.

Fig. 2(b) shows the number of unique targets and sources over time. A detailed analysis shows a relative stability of number of sources and targets. This stability in number of attackers confirms that it should be possible to predict attackers’ tactics based on past observations.

An analysis of attacked ports shows that top 10 attacked ports (with more than 10M hits) are Telnet, HTTP, SSH, DNS, FTP, BGP, Active Directory, and Nethbios ports. This shows a clear trend towards misuse of popular web services.

Predictability. Fig. 3 shows the CDF of the Shannon entropy of the different log entry elements. It helps us visualize the uncertainty about a given IP address, port number or target appearing in the logs, and thus estimate our ability to predict those values. To obtain this figure, we estimate the probability of each victim, source or port being attacked each day. For example, for each port i, we compute:

\[
Pr(\text{Port } i \text{ on day } j) = \frac{\text{Attacks on Port } i \text{ on day } j}{\text{Attacks on day } j}
\]

We also compute the entropy for each day and aggregate it overall using the CDF. Previous work [45] showed that, following Fano’s inequality, the entropy correlates with predictability. We observe that ports numbers have the lower entropy distribution, indicating a small set of targeted ports: 80% of attacks target a set of 2^{12} = 4096 ports, indicating high predictability. We also observe that victims are more predictable than sources, as 90% of victims lie within a list of 2^{14} = 16,384 sources. Victims’ set is thus significantly smaller and more predictable than attackers’ set.

Intensity. Fig. 5(a) shows the inter-arrival time of attacks in hours, and Fig. 5(b) shows the inter-arrival time of attacks in seconds. We observe that almost all attacks occur within 3-minute windows. IP addresses and /24 subnetworks have similar behavior. In particular, Fig. 5(b) shows that in short time intervals, 85% of /8 subnetworks have short attack inter-arrival time indicating the bursty attacks on such networks. Attackers target subnetworks for short time and then disappear.

Victims’ Profile. Fig. 4(a) shows the number of attacks per day on targets, with mean number of daily attacks on targets of 58.46 and median of 1. We observe three distinct victims’ profiles: (1) rarely attacked victims: 87% of targets get less than 10 attacks day, indicating many victims seldom attacked; (2) lightly attacked victims: 11% of victims get 10 to 100 attacks a day; (3) heavily attacked victims: only 2% of targets are under high attack (peaking at 11M a day). In other words, most attacks target few victims.

Attackers’ Profile. Fig. 4(b) shows the number of victims attacked by each source per day, with mean number of daily attacks of 45.85 and median of 2. We observe that 80% of sources initiate less than 10 attacks a day, indicating most sources are stealth. A small number of sources generates most attacks (up to 10M daily).
This broadly indicates two main categories of attackers: stealth and heavy hitters. In our data set, we observe that several of the top heavy attackers (more than 20M attacks) come from IP addresses owned by ISPs in the UK.

**Attacks’ Characteristics.** Fig. 4(c) shows the CDF of the number of unique sources seen by each active target a day. We focus on active victims: victims that did report an event on that particular day because, as previously discussed, many victims report attacks rarely thus creating a strong bias towards 0 otherwise. The figure contains attackers shared with other targets (common attackers) and attackers unique to a specific victim. 90% of victims are attacked by 40 unique sources and 60 shared sources. This shows that, from the victim’s perspective, targets observe more shared sources than unique ones. Compared to previous work [44, 27], this reinforces the past trend of targets having many common attackers.

Fig. 4(d) shows that 90% of sources attack 30 common victims and 60 unique victims. Although attackers share a large number of common victims, they also uniquely attack specific victims. Note that in Fig. 4(c) and Fig. 4(d), we observe again three types of victims and two types of attackers.

**Observations.** A significant proportion of victims (70% in Fig. 4(a)) contributes a single event overall. After thorough investigation, we find that these *one-time contributors* can be grouped into clusters all reporting the same IP address within close time intervals (often within one second). In other words, many contributors share only one attack event, at the same time, about the same potentially malicious IP address. Similarly, many contributors only contribute one day out of the two months (Fig. ??). We find that these contributors correlate with the aforementioned one-time contributors.

We remove victims that do not share much. In particular, we remove victims that (1) share one event overall, and (2) contribute only one day and less than less than 20 events over the two month (i.e., 10% of mean total contributions per victim 2,263). This data processing maintains properties identified in this section and reduces the number of considered victims from 800,000 to 188,522, corresponding to the removal of about 2M attacks. This filtering maintains a high diversity of contributors, and seeks to model real-world scenarios (as opposed to focusing on large contributors).

**5. PRIVACY-PRESERVING COLLABORATIVE PREDICTIVE BLACKLISTING**

We present an experimental evaluation of the SIC framework focused on (1) investigating which *select metrics* work best to estimate the benefits of sharing (measured as the resulting improvement in prediction accuracy), and (2) measuring what *merging strategies* (i.e., what data to share) provide the best privacy/accuracy trade-off. To do so, we use the DShield dataset built in Sec. 4. Experiments involve 188,522 contributing entities, each reporting an average of 2,000 attacks, for a total of 2 billion attacks.

**5.1 Experimental Setup**

Experiments are implemented in R. Source code will be released along with the final version of the paper and can already be obtained, anonymously, via the PC chairs.

**General Parameters.** For the prediction algorithm, we use a one-week window for training ($T_{\text{train}} = 7$) and aim to predict attacks
for the next day ($T_{tot} = 1$). We run the SIC framework at each time step. As previously discussed, organizations do not run SIC with all possible other organizations, but focus on a few potential partners. To model this, we take a sampling approach: For each iteration, we select 100 victims at random from the set of all 188,522 possible victims and run our select/merge algorithms. We average our results over 100 iterations.

**Select Algorithms.** We analyze how well different collaboration metrics (i.e., select strategies) perform in comparison to each other, where performance is measured in terms of resulting improvement in prediction accuracy.

SIC supports both set-based (Intersection-Size and Jaccard) and correlation-based (Pearson and Cosine) metrics. With the former, the input of each entity $V_i$ is a set of unique attacking IP addresses $S_i$. Intersection-Size returns the number of IP addresses attacking both parties, while Jaccard is the ratio between the size of set intersection and the size of the union. By contrast, for correlation to work between two entities $V_i$ and $V_j$, they need to agree on the range of IPs captured in $S_i$ and $S_j$. We assume that both parties know the global list of suspicious IP addresses. In practice, parties can agree on the range via secure computation or fetch known malicious IP address lists from repositories such as DShield.

Metrics are computed pairwise across entities. As a consequence, we obtain a matrix estimating data sharing benefits among all possible pairs. We assume that parties select partners by maximizing their potential benefits in the collaboration matrix. Typically, each party picks the list of partners with the largest potential benefits. W.l.o.g. we consider that the 50 largest collaboration pairs are selected (i.e., only 1% of 100 * 99/2 = 4,950 possible pairs as we consider 100 victims). Such a small number provides a high degree of privacy and takes a conservative stance by limiting the possible improvement in the prediction accuracy. Recall that the goal of our experimental evaluation is to understand which metrics work better, not to establish the optimal size of collaboration pools.

**Merge Algorithms.** We consider two types of algorithms, Union with Associated Data and Intersection with Associated Data (see Sec. 3.3). With the former, partners share all data known by each party prior to current time $t$ and share it with each selected partner. It is a generous strategy that enriches others’ datasets rapidly. With the latter, partners only share events from those IP addresses that belong to the intersection (i.e., that attacked both partners) and thus is a more conservative option. This approach can help reinforce knowledge about given adversaries, and thus help better predict attacks.

**Accuracy.** As discussed in Sec. 2.3, we measure the prediction success by computing the number of True Positives (TP), as in prior work [44, 52]. True positives correspond to successfully predicted attacks. Specifically, we measure the Improvement $I$ as the following fraction:

$$I = \frac{TP_c - TP}{TP}$$

where $TP$ is the number of True Positives before collaboration and $TP_c$ is the number of True Positives after collaboration. We note that improvement can be measured over all entities, or for specific entities. In the following, we give both improvement measures.

5.2 Results

**Determining the Value of $\alpha$.** Before testing the performance of select/merge algorithms, we need to identify appropriate $\alpha$ values for the EWMA prediction algorithm by evaluating the performance of the prediction. For small values of $\alpha$, the prediction algorithm aggregates past information uniformly across the training window to craft predictions. In other words, events in the far past have a similar weight to events in the short past and the algorithm has a long memory. On the contrary, with a large $\alpha$, the prediction algorithm focuses on the most recent past events; it has short memory.

Fig. 6(a) shows the evolution of the baseline prediction for different values of $\alpha$, plotting the True Positives (TP) sum of all 100 victims averaged over 100 iterations. Values between $\alpha = 0.5$ and $\alpha = 0.9$ perform best. This can be explained by remembering the “bursty nature” of web attacks, as discussed in Sec. 4. Prediction algorithms that react fast to the apparition of new attackers perform better. In the following experiments, we set $\alpha = 0.9$.

**Visualizing Predictions.** Fig. 6(b) shows a visualization of the prediction. When an attack occurs (blue square), the algorithm systematically predicts an attack (red cross) in the next time slot. Because $\alpha = 0.9$, the last attack event has a larger weight.

**Baseline Prediction.** We verify the effectiveness of the prediction algorithm by correlating the information known by targets prior to
collaboration with their ability to predict future attacks. We obtain that, as expected, targets that know more about past attacks (large $S_i$), successfully predict more future attacks. We compute the correlation $R$ and obtain $R > 0.9$ on average, which indicates strong correlation. This, once again, suggests that collaboration increases prediction success. We visualize this correlation for a specific simulation in Fig. 7(a).

5.2.1 Select Strategies

Fig. 7(b) shows the accuracy of predictions for different select methods over the course of one week, fixing the merge algorithm to Intersection with Associated Data, as it provides the strongest privacy protection. We sum the total number of TP for “collaborators” (i.e., entities that do share data) and “non-collaborators” (entities that do not share data, thus performing as in the baseline). We observe that Intersection-Size performs best, followed by Jaccard, and Cosine/Pearson. The overall decrease in sum of True Positives after day 10 is due to the decrease of attacks on those days (Fig. 2(a)).

Improvement Over Baseline. In Fig. 7(c), we compare the prediction accuracy of upper-bounds, baseline, and collaboration using Intersection-Size as the select metric and merging data using Intersection with Associated Data. We sum the total number of TP for collaborators selected by the Intersection-Size metric.

Remind that with the Global Upper Bound (GUB), every victim shares with every other victim and predicts perfectly. With the Local Upper Bound (LUB), organizations do not share anything but still predict perfectly.

The accuracy of Intersection-Size predictions tends to match LUB, showing that collaboration helps perform as well as a local perfect predictor. Note that prediction performance can be significantly improved (thus, reducing the “gap” with GUB) by enabling more collaboration pairs than the conservative 50% of all pairs considered in our experiments.

Effects of Selective Collaboration. Table 4 summarizes prediction improvements for collaborators given different select metrics, reporting the mean, max, and min improvement, as well as number of collaborators. Correlation-based metrics provide a less significant prediction improvement than set-based metrics. Mean $I$ for Pearson and Cosine is about 40%. Notably, the Intersection-Size has a 105% mean improvement. Also, mean $I$ for Jaccard is about 60%. Naturally, the improvement can also be measured for each entity: $I$ for Intersection-Size is up to 700%.

Differences between select metrics are due to several reasons. First, metrics that use a normalization factor (i.e., all but Intersection-Size) tend to create partnerships of small collaborators. By contrast, Intersection-Size leads to better performance because it promotes collaboration with larger victims. To confirm this hypothesis, we measure the set size of collaborators according to different metrics (Fig. 8) and confirm that metrics with a normalization factor tend
to pick collaborators that know less.

Second, correlation-based metrics tend to select partners that are too similar: maximum correlation values are close to 1, whereas maximum Jaccard values get to 0.5 only. Although this implies that targets learn to better defend against specific adversaries, it also leads to little acquired knowledge.

Third, depending on the select metric, at each time step, victims may partner with previous collaborators, or with new ones. We find that Intersection-Size, Pearson, and Cosine lead to stable groups of collaborators with about 90% reuse over time, whereas Jaccard has larger diversity of collaborators over time. This is because about 20% of victims have high Jaccard similarity versus only 4% for correlation-based metrics providing a larger pool of potential collaborators. Hence, if Intersection-Size helps a few learn a lot, Jaccard helps many victims over time.

Statistical Analysis. We take a closer look at Fig. 8. A t-test analysis shows that the mean of the number of events known by collaborators differs significantly ($p < 0.0005$) across all pairs of select metrics but Cosine and Pearson.

Furthermore, if one categorizes collaborators as “large” if they know more than 500 events, and “small” otherwise, and consider Cosine and Pearson as one (given the t-test result), we obtain a $3 \times 2$ table of select metrics and size categories. A $\chi^2$-test on the table shows that categorization differences are statistically significant: Intersection-Size tends to select larger collaborators, but also more collaborators than Pearson/Cosine (see Table 4). Other metrics tend to select small collaborators. We obtain $\chi^2(2, N = 448) = 191.99, p < 0.0005$, where 2 is the degrees of freedom of the $\chi^2$ estimate, and $N$ is the total number of observations.

Coalitions. Recall that, at each time step, entities can decide to partner with a number of other entities. Table 4 shows the mean, standard deviation, and median number of collaborators per party for different collaboration metrics. We observe that with Jaccard, entities tend to select less collaborators. Other metrics tend to have similar behavior and have entities to collaborate with about 5 other entities out of 100. This is in line with previous work [27], which showed the existence of small groups of correlated entities.

We also observe that, after a few days (usually 2), Intersection-Size, Pearson, and Cosine converge to a relatively stable group of collaborators. From one time-step to another, parties continue to collaborate with about 90% of entities they previously collaborated. In other words, coalitions are relatively stable over time. Comparatively, Jaccard has a larger diversity of collaborators over time.

5.2.2 Merge Algorithms

The next step is to compare the average prediction improvement $I$ for different merge algorithms. As shown in Fig. 9, Intersection with Associated Data performs almost as good as Union with Associated Data with all select strategies. Actually, it performs better with Jaccard. Merging using the union entails sharing more information, thus, one would expect it to always perform better.

However, using Union with Associated Data, we notice that organizations quickly converge to a stable set of collaborators, and obtain a potentially lower diversity of insights over time. With most metrics, the set of collaborators is stable over time anyways, and so union does perform better than intersection. But, as previously discussed, Jaccard tends to yield a larger diversity of collaborators over time and thus benefits more from Intersection with Associated Data as it re-enforces such diversity of insights.

5.3 Performance

We now estimate the operational cost of our techniques and show that it is appreciably low, thus demonstrating the viability of our approach in the real-world. Specifically, we evaluate the overhead introduced by the privacy protection layer, i.e., the privacy-preserving information sharing protocols used to securely execute the select/merge algorithms.

Excluding correlation-based metrics (due to lower accuracy improvement), the protocols for privately selecting partners (Intersection-Size and Jaccard) can be realized based on Private Set Intersection Cardinality (PSI-CA), and we choose the instantiation proposed in [13], which incur computation and communication overhead linear in sets size. Privacy-preserving merging is realized using the Private Set Intersection (PSI) with Data Transfer protocol from [14] in order to realize Intersection with Associated Data.

We implemented protocols from [13] and [14] in C, using GMP and OpenSSL cryptographic libraries, and conducted experiments on two Intel Xeon desktops with 3.10GHz CPU, connected by a 100 Mbps Ethernet link. Fig. 4(c) shows that 98% of targets are attacked by approximately 200 sources. Using sets of size 200, we find that it takes approximately 400 ms to run the PSI protocol in [14] and 550 ms to run PSI-CA in [13]. Assuming that $n$ organizations contribute to our framework, we have a total of $n \times (n - 1)$ pairwise executions.

Naturally, it is not reasonable to consider all possible partnerships in a large pool of organizations. Parties first identify a set of potential partners, such as organizations within an industry, and
Average Prediction Improvement for Collaborators

When merging datasets, organizations sharing only common attacks is almost as useful as sharing all organizations, the total computation overhead (again, assuming n = 100 partners and merging with all partners) would amount to 45 minutes for benefit estimation and 33 minutes for dataset merging, which is still reasonable for computations that are performed, e.g., once a day. Also considering that more efficient protocols for privacy-preserving information sharing are being proposed, e.g., [16], we conclude that overhead introduced by the privacy protection layer is appreciably low and does not impede the deployment and the adoption of our techniques.

5.4 Summary of Results

Knowing More Means Predicting More. Our experiments show that targets that know more tend to predict more attacks. This confirms our hypothesis about the opportunity to collaborate with targets exposed to numerous attacks. However, the simple “more-data-the-better” approach conflicts with the basic privacy principle of data minimization. Thus, the challenge consists in identifying partners that help most. Choosing partners based on higher values of Intersection-Size (i.e., number of common attacks) works best and provides convenient privacy properties since it only discloses information about attackers entities already know of.

Sometimes Sharing Does Not Help Much. In some cases, data sharing does not yield significant improvements. Our analysis reveals that differences in the definition of similarity can lead to significant variations in prediction accuracy. We find that, when considering correlation-based similarity between victims’ profiles, small contributors are paired together, leading to small overall improvements. By contrast, set-based metrics favor larger contributors and thus yield larger overall improvement.

Sharing Only Common Attacks Is Almost As Useful As Sharing Everything. When merging datasets, organizations sharing only information about common attacks (i.e., using Intersection with Associated Data) achieve a good trade-off between privacy and utility as the improvement is almost as good as when sharing everything.

Intuitively, merging using intersection helps because it reinforces knowledge of a particular attacker, while using union helps victims attacked by varying group of attackers. Hence, we find that victims benefit as much from improving their knowledge of current attackers, as learning about sources that attack them next. In other words, learning information about attackers that targeted a victim in the past is similar to learning about attackers that might target a victim in the future.

5.5 Limitations

The DShield dataset used in our experiments is biased toward small organizations voluntarily reporting data, thus it might not be directly evident how to generalize our results. However, our findings show strong statistical evidence that collaboration metrics affect data sharing performance in interesting ways. Our proposed algorithms and methodology can serve as the basis for further experiments that explore the concept of privacy-friendly data sharing.

As in previous work [44, 52], we did not consider false positives and focused on measuring the prediction algorithm’s TP. As discussed in Sec. 2.3, this is reasonable as the prediction algorithm identifies suspicious IP addresses that deserve further scrutiny and that are subsequently processed by blacklisting algorithms, which actually block sources, even though false positives might increase the computational load and complexity of the blacklisting algorithm by providing larger inputs.

Finally, note that this paper does not aim to present a finished product, but to demonstrate the viability and effectiveness of privacy-enhancing technologies on collaborative threat mitigation. While the overhead introduced by our peer-to-peer approach is still non-negligible, it is significantly lower than existing alternatives such as FHE. Also, a few improvements could be explored in future work to improve performance, including parallelization, centralization, and/or sampling.

6. RELATED WORK

6.1 Collaborative Security Initiatives

Public Sector. In 1998, U.S. President Clinton initiated a national program on Critical Infrastructure Protection [7], which promoted collaboration between government and private sector, and created the Financial Sector Information Sharing and Analysis Center (FS-ISAC). In 2003, this was extended to virtual systems and IT infrastructures with the Homeland Security Presidential Directive 7 (HSPD-7), and recently reinforced [35]. In 2013, the US House of Representatives passed the Cyber Intelligence Sharing and Protection Act (CISA), which met huge opposition as it granted broad immunity to data sharing entities, and took generous views on what data could be shared and with whom. The bill was not voted on by the Senate and the debate is still ongoing with similar proposals [48]. Standardization bodies also push collaborative frameworks and established appropriate data formats (IDMEF, IODEF RFC 5070 [12]), collaboration protocols (the Real-time Inter-network Defense RFC 6545 [33]), and guidelines (ISO 27010, ITU-T SG17).

Private Sector. The RedSky Alliance [41] helps security professionals share intelligence after a vetting process for trust establishment. Another example is TF-CSIRT (Task Force of Computer Security Incident Response Teams) [46], which improves coordination among European Community Emergency Response Teams (CERTs). Besides DShield [42], other community-based initiatives focus on sharing and correlating security data. DOMINO (Distributed Overlay for Monitoring InterNet Outbreaks) [51] provides

Figure 9: Number of True Positives (TP) for two different merge algorithms: Union/Intersection with Associated Data.
distributed intrusion detection promoting collaboration among nodes. In Europe, the Worldwide Observatory of Malicious Behaviors and Attack Threats (WOMBAT) gathers security related data in real-time. Symantec also introduced a data sharing platform, WINE. Finally, the MITRE Corporation \cite{32} developed file formats (STIX), collaboration protocols (TAXII), and repository formats (CAPEC, MAEC) for structure threat information exchange.

**Barriers to Adoption.** These initiatives have had little success, as pointed out by the Federal Communications Commission’s Working Group on Communications Security, Reliability and Interoperability Council’s (CSRIC) \cite{9}. Existing solutions rely on manual out-of-band channels to establish trust. For instance, the RedSky alliance relies on a long and costly vetting process that requires manual labor to verify the trustworthiness of potential partners. Furthermore, organizations need to reveal their datasets to a centralized third-party and rely on it to for security. Thus, they have limited control over how their data is shared with other participants.

Existing solutions have a turnover of a few days for RedSky alliance, to a few weeks in the case of ISACs. In other words, the feedback is significantly slower than the spread of malware. It is difficult for companies to quantify how much others are contributing, and this lack of transparency discourages contributions. By contrast, our solutions offer privacy-preserving benefit measures to provide immediate feedback, real-time turnover, and transparency of contributions.

**6.2 Collaborative Threat Mitigation**

Previous research work \cite{27, 44, 38, 52} has relied on central repositories for collaborative threat mitigation without providing privacy protection. Katti et al. \cite{27} showed that correlated attacks, i.e., attacks mounted by same IP sources against different victims, are prevalent on the Internet. Pouget et al. \cite{38} obtained similar results using distributed honeypots for observation of malicious online activities. Katti et al. further clustered victims that share common attacks and found that (1) correlations among victims are persistent over time; and (2) collaboration among victims from correlated attacks dramatically improves malicious IP detection time. This contrasts with our results and shows that similarity metrics must be carefully defined for data sharing to improve prediction.

Zhang et al. \cite{52} introduced the concept of predictive blacklisting and suggested that victims should predict future attackers based on their logs and those of other similar victims. Sildo et al. \cite{44} aim to forecast attack sources based on past, shared attack logs using an implicit recommendation system (as opposed to link-analysis in \cite{52}) and improved prediction accuracy as well as robustness against poisoning attacks.

**6.3 Privacy-Preserving Data Sharing**

The security research community has attempted to balance data utility with privacy protection for data sharing via anonymization. Lincoln et al. \cite{31} suggested sharing sanitized security data for collaborative analysis of security threats. Specifically, they removed sensitive data prior to sharing, such as IP addresses. Other mechanisms were proposed to anonymize traces, ranging from prefix-preserving anonymization of IP addresses \cite{43, 49} to statistical obfuscation \cite{1}. However, previous work showed that inference attacks can de-anonymize network traces \cite{8}, and that it is difficult to maintain data utility \cite{28, 30, 34, 47}. Our work complements anonymized data sharing efforts since in the merge phase, parties could decide to share data sanitized according to existing proposals.

Alternatively, Applebaum et al. \cite{3} presented protocols for private data aggregation, and envisioned, as a use case, the analysis of aggregated data for anomaly detection. Their approach requires a semi-trusted proxy aggregator and only provides participants with aggregated counts of common data points across multiple entities.

Burkhart et al. \cite{6} proposed a distributed solution based on secure multi-party computation and secret sharing that supports aggregation of security alerts and traffic measurements among peers, e.g., to estimate global traffic volume. These protocols are secure as long as the majority of peers do not collude, assume a reliable infrastructure to distribute and store the key shares, and incur a large number of rounds and communication overhead.

Aggregation helps detect malicious attacks or compute traffic statistics but, alas, it operates similar to large alert repositories: it identifies most prolific attack sources and yields global models. As shown in \cite{44, 52}, generic attack models miss a significant number of attacks, especially when attackers choose targets more strategically, focusing on a few known vulnerable networks.

**7. CONCLUSION**

This paper presented a novel privacy-friendly approach to collaborative threat mitigation. We showed how organizations can quantify expected benefits in a privacy-preserving way (i.e., without disclosing their datasets) before deciding whether or not to collaborate. Based on these benefits, they can then organize into coalitions and decide what/how much to share. We focused on collaborative predictive blacklisting, evaluated our techniques on a real-world dataset, and observed a significant improvement in prediction accuracy (up to 105%, even when only 1% of all possible partners collaborate).

Our first-of-its-kind analysis shows that certain collaboration strategies work better than others. The number of common attacks provides a good estimation of the benefits of sharing, as it drives entities to partner with more knowledgeable collaborators. Also, sharing information only about common attacks proves to be almost as useful as sharing everything. This suggests that victims benefit as much from improving their knowledge about entities that currently attack them, as from learning about entities that do not attack them now, but might in the future.

We demonstrated the benefits of privacy-preserving information sharing on collaborative threat mitigation and established that data sharing does not have to be an “all-or-nothing” process: by relying on efficient secure computation, it is possible to only share relevant data, and only when beneficial. Privately assessing whether or not, and how, entities should partner up prompts interesting challenges, which our work is the first to tackle.

Finally, we hope that these insights can inform the ongoing debates in the US Congress regarding the Cyber Intelligence Sharing and Protection Act (CISPA) \cite{48}. As part of future work, we intend to study other metrics for partner selection (e.g., dissimilarity) and experiment with other prediction algorithms and incentive mechanisms. We will also explore how to adapt our approach to other collaborative security problems, e.g., spam filtering \cite{11}, virus detection \cite{21}, or DDoS mitigation \cite{36}.

**8. REFERENCES**

\[1\] E. Adar, User 4xxxxx9: Anonymizing query logs. In Query Log Analysis Workshop, 2007.

\[2\] R. Agrawal, A. Evfimievski, and R. Srikant. Information sharing across private databases. In SIGMOD, 2003.

\[3\] B. Applebaum, H. Ringberg, M. Freedman, M. Caesar, and J. Rexford. Collaborative, privacy-preserving data aggregation at scale. In PETS, 2010.

\[4\] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko, The one-more-RSA-inversion problem and the security of Chaum’s blind signature scheme. Journal of Cryptology, 16(3), 2003.
