Releasing ARP Data with Differential Privacy Guarantees For LAN Anomaly Detection

Norrathep Rattanavipanon[1], Donlapark Ponnoprat*[2], Hideya Ochiai[3], Kuljaree Tantayakul[1],
Touchai Angchuan[4], and Sinchai Kamolphiwong[4]

[1]College of Computing, Prince of Songkla University
{norrathep.r, kuljaree.t}@phuket.psu.ac.th
[2]Department of Statistics, Chiang Mai University
donlapark.p@cmu.ac.th
[3]Graduate School of Information Science and Technology, The University of Tokyo
jo2lxq@hongo.wide.ad.jp
[4]Faculty of Engineering, Prince of Songkla University
{touch, ksinchai}@coe.psu.ac.th

Abstract—**Anomaly detection has emerged as a popular technique for detecting malicious behaviors in local area networks (LANs). Various aspects of LAN anomaly detection have been widely studied. Nonetheless, the privacy concern about individual users or their relationship in LAN has not been thoroughly explored in the prior work. In some realistic cases, the anomaly detection analysis needs to be carried out by an external party, located outside the LAN. Thus, it is important for the LAN admin to release LAN data to this party in a private way in order to reveal no information about LAN users; at the same time, the released data must also preserve the utility of being able to detect anomalies. This paper investigates the possibility of privately releasing ARP data that can later be used to identify anomalies in LAN. We present two approaches and show that they satisfy different levels of differential privacy – a rigorous and provable notion for quantifying privacy loss in a system. Our real-world experimental results confirm practical feasibility of our approaches. With a proper privacy budget, both of our approaches preserve more than 90% utility of the released data.**

I. Introduction

Security of local area networks (LANs) has been getting more attention in the last few decades. Traditional LAN defense mechanisms based on a firewall are no longer effective in preventing malware infection since malware can circumvent the firewall and infect the network through other means. A prominent example is the recent emergence of ransomware that can infect LAN devices via phishing attacks; these attacks remain effective even if the LAN's firewall is active and configured correctly. In addition, with the rise of the Internet-of-things (IoT), the so-called "smart" devices have become widely popular and, at the same time, are also extremely vulnerable to malware attacks. These devices may be infected from the outside world and introduce malware to the LAN.

Several anomaly detection techniques have been proposed to detect malicious activities in LAN. Among those, techniques based on the Address Resolution Protocol (ARP) are shown to be promising in detecting LAN anomalous behaviors without requiring a change to existing devices [6], [11], making them suitable for the current IoT networks.

Despite this success, there still remains a severe privacy concern to LAN users, which has not been explored in the previous work. Often times, the anomaly detection must be performed by an entity outside of LAN or third-party software. Thus, it is equally important to ensure privacy of the data exposed to this external, potentially malicious, entity. For instance, a LAN admin in an enterprise may choose to outsource an anomaly detection analysis to an external widely-popular service, e.g., [7], or the admin simply wants to publicly release some features of network data for transparency or academic purposes. In either case, it would require the LAN admin to output network data (which is an input to the anomaly detection algorithm) to an untrusted party. Doing so may lead to having such party learn privacy-sensitive information about the LAN users, e.g., presence of a specific LAN user.

While it is possible to simply erase all user's sensitive information (e.g., IP/MAC addresses) from the output data, this kind of technique does not provide strong and provable privacy guarantees. A motivated adversary may still be able to deanonymize users through other means, e.g., via a side-channel [8] analysis. Therefore, there is a need for a technique with *rigorous* privacy guarantees, while preserving the utility of detecting anomalies in the LAN environment.

In this paper, we present two approaches to releasing ARP data for LAN anomaly detection, with mathematically proven privacy guarantees. The first approach guarantees a weak definition of differential privacy [1], allowing to be used in scenarios where a strong privacy guarantee is not required; whereas, the second approach requires a slightly higher value of a privacy budget parameter (ε) in order to attain stronger

*Corresponding author

978-0-7381-1127-8/21/$31.00 ©2021 IEEE

privacy protection. We study the practicality of our approaches on a real-world dataset, which confirms that with a proper privacy budget, both of our approaches preserve the utility by more than 90%.

We envision that our approaches will be especially useful in a scenario where ARP data collected from a large-scale network needs to be made publicly available. As a real-world example, one of the on-going ASEAN IVO projects, called ASEAN-Wide Cyber-Security Research Testbed project, aims to: (1) capture network data from LANs across the ASEAN region, (2) determine malware behaviors based on the captured data and (3) make the captured data sharable in the public domain. Our approaches will greatly contribute to this project as a mechanism to ensure privacy protection of the released ARP data captured in the ASEAN region.

II. Related Work

To the best of our knowledge, there has been no prior work proposing a method for releasing ARP data that can be used for anomaly detection with differential privacy guarantees in the LAN setting. However, some previous work has been done under different conditions or different settings, e.g., social network [9], web browsing [3], or syndrome surveillance [4].

There are a number of existing research that aims to detect anomalies in LAN *without* providing privacy protection. The work in [12] proposes a framework to monitor a network traffic and detect anomalies in the Wireless LAN (WLAN) environment via the IEEE 802.11 MAC protocol. Nonetheless, this approach is specific to wireless LAN and thus cannot be directly applied to the wired LAN setting. Our approaches are based on ARP requests, making them suitable for both wired and wireless LAN environments.

Several prior work focuses on detecting LAN anomalies based on ARP-related data. Whyte et al. [10] propose an anomaly detection approach that distinguishes anomalous behaviors through statistical analyses of ARP traffic. Yasami et al. [11] propose to model normal ARP traffic behaviors using Hidden Markov Model (HMM). Matsufuji et al. [6] present an anomaly detection algorithm based on the degree of destination in ARP requests.

III. Preliminaries

A. Problem Statement

In this work, we consider a setting in which an entity, called Admin, possesses a LAN consisting of n User-s (i.e., computing devices). In addition, Admin introduces a monitoring device to this LAN in order to observe ARP requests of all User-s. We denote V_{jk} to be aggregate ARP requests originated from User k, measured and accumulated at the j^{th} interval. In this work, we assume the time interval to be in a unit of "a week" but our approaches are also applicable for any fixed interval (e.g., an hour, a day, a month, etc.). V_j is denoted the result after appending all ARP requests of all User-s generated in week j, i.e. $V_j = \{V_{j1}, V_{j2}, ..., V_{jn}\}$.

As shown in Figure 1, our target scenario starts by having the monitoring node (periodically) send aggregate ARP requests – $V = \{V_1, ..., V_t\}$ – to Admin, corresponding to step ❶ in Figure 1. Admin is interested in learning whether the LAN *as a whole* has had any anomalous behaviors for the last t weeks **in a private way**. Thus, in step ❷, he proceeds to apply a certain algorithm with the goal of privatizing V and then releases the result D to an external entity Analyst. In step ❸, Analyst in turn performs an anomaly detection analysis on D and returns the result O back to Admin. O contains O_i that allows Admin to identify whether the LAN contains an anomaly at week i.

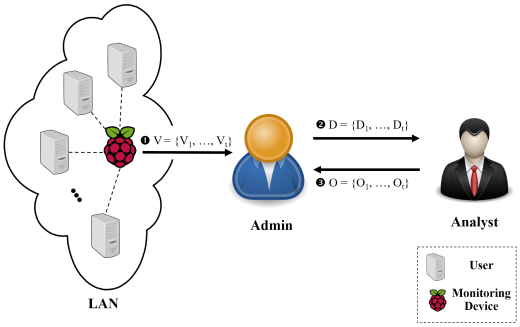

Fig. 1. Scenario considered in this work

Threat Model: Analyst is assumed to be honest-but-curious, i.e, he always honestly applies an anomaly detection algorithm on any given input data and returns the correct output to Admin. However, during the process, he may want to learn sensitive information about User-s or their relationship, and use it for his own benefits.

Goal & Scope: The goal of this work is to design approaches that can be appropriately used in step ❷ of Figure 1. In other words, our approaches must allow the process of releasing ARP data with some levels of provable privacy guarantees. Besides privacy, utility of the privatized/released data for anomaly detection is also important. We must ensure that the privatized value does not change by a significant amount, compared to the non-privatized counterpart; otherwise, it will not be useful in detecting anomalies. However, exact algorithms for determining anomalies (i.e., in step ❸) are out of scope.

B. Differential Privacy

Consider a setting in which there are n users who send individual data to a trusted curator. The curator then applies an algorithm $\mathcal{M}$ and outputs these results to an untrusted party. In a strong notion of privacy, the data of an individual must be kept private from strong adversaries – even ones who get a hand on the data of the other users.

The *differential privacy* (DP) is a probabilistic viewpoint of this notion given in a seminal paper by Dwork, McSherry, Nissim, and Smith [1]. First, we say that two datasets X and X' are *neighboring* if they differ by exactly one entry. The differential privacy is then satisfied if changing X to X' does not change the probability of observing an output of $\mathcal{M}$ by very much.

Definition 1 (Differential Privacy)**.** An algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ satisfies ε-*differential privacy* (ε-DP) if, for every pair of neighboring datasets X and X' and every subset $S \in \mathcal{Y}$,

$$\mathbb{P}\left(\mathcal{M}(X) \in S\right) \leq e^{\varepsilon}\mathbb{P}\left(\mathcal{M}(X') \in S\right).$$

where ε is referred as a privacy budget. Intuitively, a smaller value of ε leads to a stronger privacy guarantee. Conversely, a higher value of ε implies a weaker guarantee with possibly better utility/accuracy of the released data.

One useful property is the preservation of differential privacy under post-processing.

Proposition 1 (Post-processing [2])**.** For any ε-DP algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ and arbitrary random function $f : \mathcal{Y} \to \mathcal{Z}$, the algorithm $f \circ \mathcal{M}$ is ε-DP.

To introduce one of the most ubiquitous ε-DP algorithm, we start with the ℓ_1-*sensitivity* of a randomized algorithm $\mathcal{M} : \mathcal{X} \to \mathbb{R}^k$, which is the maximum ℓ_1 change in the output as a result of modifying a single datum. We denote this sensitivity as $\Delta^{\mathcal{M}}$, and formally define it as:

$$\Delta^{\mathcal{M}} = \max_{\text{neighbor } X,X'} \|\mathcal{M}(X) - \mathcal{M}(X')\|_1.$$

Theorem 1 (Laplace mechanism [2])**.** *Let $\mathcal{M} : \mathcal{X} \to \mathbb{R}^k$ be an algorithm with sensitivity $\Delta^{\mathcal{M}}$ and Y_i be a noise generated by sampling from a Laplace distribution with scale $= \Delta^{\mathcal{M}}/\varepsilon$, i.e., $Y_i \sim$ Laplace$(\Delta^{\mathcal{M}}/\varepsilon)$, then the randomized algorithm $\mathcal{A}$ defined by*

$$\mathcal{A}(X) = \mathcal{M}(X) + (Y_1, \ldots, Y_k)$$

is ε-DP.

C. Differential Privacy of ARP-Request data

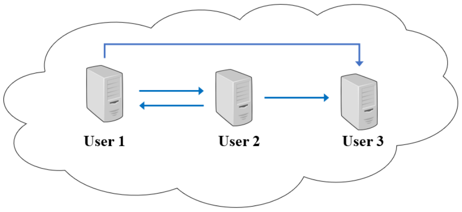

Fig. 2. Illustration of a LAN with 3 User-s and 4 ARP requests (arrows).

To understand privacy (i.e., what *concrete* information needs to be private and hidden from Analyst) in our target scenario, we first describe the characteristic of ARP-request data. Figure 2 illustrates an example of a LAN that consists of 3 User-s producing 4 ARP requests over a specific time interval. We define the (ARP-request) "degree" of User j as the number of User-s that receive ARP requests from User j. In the above example, that the degrees of User 1, 2 and 3 are 2, 2 and 0, respectively. Using this definition, we can view V_j – aggregate ARP-request data at week j – as a directed graph, where User can be represented by a node; whereas an arrow (or a directed edge) from node s to node r indicates ARP request(s) generated by User s and sent to User r. The degree of User j is then equivalent to the number of directed edges originating from User j.

Algorithm 1: Naïve Approach

```
Input: $V = \{V_1, V_2, ..., V_t\}$, $t$, $\varepsilon$
Output: $D = \{D_1, D_2, ..., D_t\}$
for $j = 1$ to $t$ do
    $D_j \leftarrow \text{SUM}(\text{DEGREE}(V_j))$
    $D_j \leftarrow D_j + \text{Laplace}(t/\varepsilon)$
end
```

As a directed graph, V_j can not be used to represent a database entry, required by Definition 1. Thus, the aforementioned notion of differential privacy does not accurately capture the privacy guarantee in our scenario. Fortunately, there has been a prior work focusing on expressing differential privacy of a graph database. Specifically, the work in [5] presents notions of differential privacy between graphs by first defining two types of neighboring graphs: two graphs are *edge-neighboring* if they differ by a single edge, and they are *node-neighboring* if they differ by a single node.

Definition 2 (Edge-Differential Privacy [5])**.** Let $\mathcal{G}$ be the set of graphs between users. An algorithm $\mathcal{M} : \mathcal{G} \to \mathcal{Y}$ satisfies ε-edge-differential privacy (ε-edge-DP) if, for every pair of edge-neighboring graphs G and G' and every subset $S \subseteq \mathcal{Y}$,

$$\mathbb{P}\left(\mathcal{M}(G) \in S\right) \leq e^{\varepsilon}\mathbb{P}\left(\mathcal{M}(G') \in S\right).$$

Since an edge in our scenario refers to ARP requests between a pair of User-s, Definition 2 provides privacy protection for these ARP requests. This means that an algorithm satisfying ε-edge-DP is guaranteed to reveal no information about all ARP requests exchanged between any pair of User-s, resulting in hiding the ARP relationship of all User-s. This, for example, could hide the source of infection in LAN as it is common for malware to use ARP as the first step to discover and infect other User-s.

Nonetheless, the guarantee provided by Definition 2 is not strong enough to provide privacy of individual User-s. To achieve this guarantee, we adopt the following definition:

Definition 3 (Node-Differential Privacy [5])**.** Let $\mathcal{G}$ be the set of graphs between users. An algorithm $\mathcal{M} : \mathcal{G} \to \mathcal{Y}$ satisfies ε-node-differential privacy (ε-node-DP) if, for every pair of node-neighboring graphs G and G' and every subset $S \subseteq \mathcal{Y}$,

$$\mathbb{P}\left(\mathcal{M}(G) \in S\right) \leq e^{\varepsilon}\mathbb{P}\left(\mathcal{M}(G') \in S\right).$$

Indeed, by removing a node we also have to remove all of its edges. One then has that ε-node-DP is stronger than ε-edge-DP. In our scenario, an algorithm satisfying ε-node-DP guarantees no information leakage about presence or absence of any individual User.

IV. Naïve Approach

The naïve approach is summarized in Algorithm 1. In the rest of this section, we discuss non-trivial details of this approach and show that it indeed satisfies ε-edge DP.

Let $V_j \in \mathcal{G}$ be the directed graph of ARP requests in week j. Let $\mathcal{M}$ be the algorithm that computes the weekly total degrees and $D_j = \mathcal{M}(V_j)$ (Line 2 of Algorithm 1), which also corresponds to the total number of edges in V_j. To preserve ε-edge-DP of each User's ARP requests, one can simply use the Laplace mechanism. To do so, we need to find an upper bound of the sensitivity $\Delta^{\mathcal{M}}$. Let V_j' be an edge-neighboring graph of V_j in week j and $D_j' = \mathcal{M}(V_j')$. Then, $|D_j - D_j'| \leq 1$ and we have

$$\Delta^{\mathcal{M}} \leq \sum_{j=1}^{t} |D_j - D_j'| \leq t.$$

Therefore, the ε-node DP is guaranteed under the following Laplace mechanism $\mathcal{A}$:

$$\mathcal{A}(V_j) = D_j + Y_j,$$

where $Y_j \sim \text{Laplace}(t/\varepsilon)$ (Line 3). To prevent excessive information loss, one needs the Laplace noise to be smaller than D_j, i.e., $t/\varepsilon < \mathbb{E}[D_j]$ or $\varepsilon > t/\mathbb{E}[D_j]$. This can be achieved in realistic settings, e.g., $\varepsilon = 2$ in our experiment (Section VI) where $t = 30$ and the lower quartile of D_j is 20.

On the other hand, a similar analysis for the ε-node-DP results in much bigger Laplace noises; consider two node-neighboring directed graphs V_j, V_j' of n users. The degrees D_j, D_j' defined as above satisfy $|D_j - D_j'| \leq n$, which cannot be improved further. Thus, in order to employ the Laplace mechanism, the noises have to be sampled from $\text{Laplace}(tn/\varepsilon)$. In contrast to the edge-DP regime, the scale of the noise comes with a factor of n. As a result, for a large number of User-s, it is no longer feasible to preserve both privacy and utility at the same time.

V. Histogram-based Approach

As seen in the previous section, the naïve approach can not be used to satisfy ε-node-DP in practice due to its high sensitivity, leading to too strong added noises which in turn significantly lowering utility of the released data. Instead, we propose a second approach utilizing a histogram that helps reduce the ε-node-DP sensitivity to a reasonable amount.

Our approach is shown in Algorithm 2. The rationale behind this approach is to transform the degree data in such a way that the sensitivity is minimized when any User is removed from V_j. Naturally, a histogram is a good fit for this approach since it provides a way to partition data into discrete groups/bins, where each bin in this case represents a range of degrees. Thus, this approach first computes the degrees of each User in a specific week and uses this degree data to construct a histogram, as shown in Line 2 of Algorithm 2. This histogram data minimizes the ε-node-DP sensitivity because removing a User from the histogram data affects only one bin, i.e., the one this User belongs, and it only decreases its bin count by one; *other histogram bins are unaffected by this change.* We then can apply the Laplace mechanism on each bin (Line 3-5). Finally, we compute and release the lower bound of the sum of degrees from the noisy histogram in Line 8.

Algorithm 2: Histogram-based Approach

```
  Input: $V = \{V_1, V_2, ..., V_t\}$, $t$, $\varepsilon$
  Output: $D = \{D_1, D_2, ..., D_t\}$
1 for $j = 1$ to $t$ do
2     $H_j \leftarrow \text{Histogram}(\text{Degree}(V_j))$
3     foreach $bin \in H_j$ do
4         $bin.count \leftarrow bin.count + \text{Laplace}(t/\varepsilon)$
5     end
6 end
7 for $j = 1$ to $t$ do
8     $D_j \leftarrow \sum_{bin \in H_j} (bin.count \times bin.low)$
9 end
```

We now formally show that the histogram-based approach satisfies ε-node-DP.

Theorem 2. *The histogram-based approach as described in Algorithm 2 is ε-node-DP.*

Proof. Let V_j and V_j' be node-neighboring directed graph at time j, i.e., V_j' can be obtained from V_j by adding or removing a single node. Let $\mathcal{M} : \mathcal{G} \to \mathbb{R}^k$ be the algorithm that computes the histogram of the degrees, i.e., the entries of $\mathcal{M}(V_j)$ and $\mathcal{M}(V_j')$ are the count of nodes by their degrees. Then $\mathcal{M}(V_j)$ and $\mathcal{M}(V_j')$ differ by one in the entry corresponding to the degree of User j, who only exists in either V_j or V_j'. Therefore, $|\mathcal{M}(V) - \mathcal{M}(V_j')| \leq 1$. It follows that

$$\Delta^{\mathcal{M}} \leq \sum_{j=1}^{t} |\mathcal{M}(V) - \mathcal{M}(V_j')| \leq t.$$

Observe that the first **for** loop (Line 1-6) in Algorithm 2 can be written as a randomized algorithm $\mathcal{A} : \mathcal{G} \to \mathbb{R}^k$ defined by

$$\mathcal{A}(G) = \mathcal{M}(G) + (Y_1, \ldots, Y_k),$$

where $Y_i \sim \text{Laplace}(t/\varepsilon)$. It follows from Theorem 1 that $\mathcal{A}$ is ε-node-DP.

Then, we denote the second **for** loop (Line 7-9) in Algorithm 2 as f. By Proposition 1, we have the histogram-based approach, which is defined as $f \circ \mathcal{A}$, is also ε-node-DP.

□

VI. Evaluation

A. Experimental Setup

To assess utility of the proposed approaches, we collected ARP requests generated by our lab network from August 9, 2019 to March 6, 2020. As mentioned in Section III-A, this collection is performed by introducing a small monitoring device to our LAN. We implement the monitoring device atop of a raspberry Pi 3B.

In the context of differential privacy, the utility metric is generally defined as a relative error between the released privatized values z and the non-privatized aggregates z^*. We adopt a similar approach and select the root-mean-square error (RMSE) as our utility metric:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{z[i]^{*} - z[i]}{z^{*}[i]}\right)^{2}}$$

where $z[i]$ and $z^{*}[i]$ represent the i^{th} data-point in z and z^{*}, respectively. For our proposed approaches, $z^{*}[i]$ corresponds to the (estimated) sum of all User's ARP degrees observed in week i, while $z[i]$ refers to the privatized output on the same ARP data.

B. Parameter Selection

As we collected ARP requests from 328 User-s over a 30-week period, $t = 30, n = 328$. The naïve approach involves with no other parameters. Meanwhile, the histogram-based approach consists of two additional parameters: the number of bins and the width of each bin. Intuitively, a larger number of bins leads to smaller bin counts. In such case, the Laplace noise injected by the histogram-based approach would become too large, severely hurting utility of the released data.

To avoid this problem, we select the number of histogram bins to be a relatively small number: 3. Specifically, we choose the first two bins to correspond to the number of User-s whose degree is 1 and 2, respectively; the third bin contains the number of User-s with degree ≥ 3.

C. Results

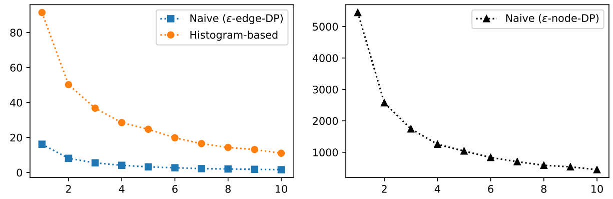

Fig. 3. RMSE in % (y-axis) vs ε values (x-axis), avg over 100 runs.

Figure 3 shows the impact of ε on the utility of the proposed approaches. For the naïve approach, to preserve $> 90\%$ utility, ε can be as low as 2. On the other hand, ε in the histogram-based approach must be ≥ 10 in order to retain the same amount of utility. A higher value of ε is needed in the second approach in order to satisfy the stronger privacy guarantee, i.e., ε-node DP; whereas, the naïve approach can not realistically support ε-node DP. The same figure also shows the error rate when we adapt the naïve approach to support ε-node-DP by increasing the sensitivity from t to tn. As expected, the error becomes extremely large, limiting the utility of the released data; at the same value of ε, RMSE of the naïve approach with ε-node-DP is > 40 times larger than that of the histogram-based approach.

Figure 4 illustrates the comparison between the released (privatized) data by the histogram-based approach and the non-privatized aggregates. Their RMSE is around 10%, resulting in $\approx 90\%$ utility. Despite a slight difference in value, the released data still preserves the same pattern as the non-privatized aggregates. As seen in Figure 4, both types of data indicate a spike at week 8, which may be used to conclude an existence of an anomaly.

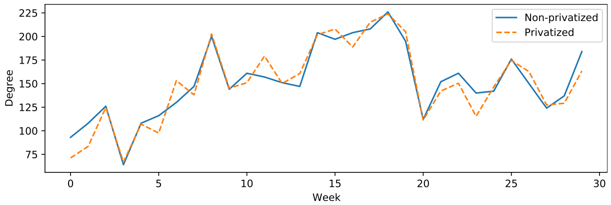

Fig. 4. Original (non-privatized) data vs released (privatized) data produced by the histogram-based approach with $\varepsilon = 10$

VII. Conclusion

This paper presents two approaches to privately releasing ARP-request data that can later be used for identifying anomalies in LAN. We prove that the naïve approach satisfies edge-differential privacy, and thus provides privacy protection on the user-relationship level. On the other hand, the histogram-based approach comes with a more expensive privacy budget but can provide node-differential privacy, thus leaking no information about a presence of each individual user. Feasibility of our approaches is demonstrated via a real-world experiment; both of our approaches are shown to preserve more than 90% utility of the released data.

Acknowledgments: The ASEAN IVO (http://www.nict.go.jp/en/asean_ivo/index.html) project, ASEAN-Wide Cyber-Security Research Testbed Project, was involved in the production of the contents of this work and financially supported by NICT (http://www.nict.go.jp/en/index.html). This work is also financially supported by Chiang Mai University.

References

[1] Cynthia Dwork, Frank Mcsherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, 2006.

[2] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(34):211–407, 2014.

[3] Liyue Fan, Luca Bonomi, Li Xiong, and Vaidy Sunderam. Monitoring web browsing behavior with differential privacy. In *WWW*, 2014.

[4] Liyue Fan and Li Xiong. Differentially private anomaly detection with a case study on epidemic outbreak detection. In *ICDM*. IEEE, 2013.

[5] Michael Hay, Chao Li, Gerome Miklau, and David D. Jensen. Accurate estimation of the degree distribution of private networks. *ICDM*, 2009.

[6] Kai Matsufuji, Satoru Kobayashi, Hiroshi Esaki, and Hideya Ochiai. Arp request trend fitting for detecting malicious activity in lan. In *ICUIMC*, 2019.

[7] Microsoft Azure. Anomaly detector, 2020.

[8] Mudhakar Srivatsa and Mike Hicks. Deanonymizing mobility traces: Using social network as a side-channel. In *CCS*, 2012.

[9] Qian Wang, Yan Zhang, Xiao Lu, Zhibo Wang, Zhan Qin, and Kui Ren. Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy. *IEEE Transactions on Dependable and Secure Computing*, 15(4):591–606, 2016.

[10] David Whyte, EVANGELOS Kranakis, and P Van Oorschot. Arp-based detection of scanning worms within an enterprise network. In *ACSAC*, 2005.

[11] Yasser Yasami, Majid Farahmand, and Vahid Zargari. An arp-based anomaly detection algorithm using hidden markov model in enterprise networks. In *ICSNC 2007*. IEEE, 2007.

[12] Jihwang Yeo, Moustafa Youssef, and Ashok Agrawala. A framework for wireless lan monitoring and its applications. In *Proceedings of the 3rd ACM workshop on Wireless security*, 2004.