Hindawi
Security and Communication Networks
Volume 2022, Article ID 1403200, 15 pages
https://doi.org/10.1155/2022/1403200

Research Article

Detecting Anomalous LAN Activities under Differential Privacy

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, Phuket 83120, Thailand*
[2]*Data Science Research Center, Department of Statistics, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand*
[3]*Graduate School of Information Science and Technology, The University of Tokyo, Tokyo 113-8656, Japan*
[4]*Faculty of Engineering, Prince of Songkla University, Songkhla 90110, Thailand*

Correspondence should be addressed to Donlapark Ponnoprat; donlapark.p@cmu.ac.th

Received 5 October 2021; Revised 11 January 2022; Accepted 27 January 2022; Published 12 April 2022

Academic Editor: George Drosatos

Anomaly detection has emerged as a popular technique for detecting malicious activities 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 protect privacy of 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 four approaches, namely, naïve, histogram-based, naïve-δ, and histogram-based-δ 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, all of our approaches preserve more than 75% utility of detecting anomalies in the released data.

1. 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 simply circumvent the firewall or infect the network through other means [2, 3]. 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 [4, 5]. 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 [6]. These devices may be infected from the outside world and introduce malware to the LAN.

To overcome this challenge, 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 anomalous activities in LAN without requiring a change to existing devices [7, 8], making it suitable to the current IoT networks.

Despite this success, there still remains a severe privacy concern to LAN users, which has not been thoroughly explored in the previous work. Often times, the anomaly detection must be performed by an entity outside LAN [9–11] or third-party software [12, 13]. Thus, it is equally important to ensure privacy of the data exposed to this external and 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., Microsoft's Anomaly Detector [12], or the admin simply wants to 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. For example, it may directly disclose personally identifiable information (PII), e.g., IP/MAC addresses, which can be used to uncover the identity of LAN users. It may also cause an indirect information leakage by revealing information about access patterns (e.g., the time of the day that a specific user is online) or relationship between users [14].

While it is possible to simply erase all users' sensitive information 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., performing a side-channel analysis [15] or correlating the remaining network traces with the physical world data [16]. Therefore, there is a need for a technique with *rigorous* privacy guarantees, while preserving the utility of detecting anomalies in the LAN environment.

Contributions: to this end, the goal of this paper is to investigate the possibility of privately publishing ARP data that can later be used to identify anomalies in LAN. Our work presents the following contributions:

(i) *Privacy Notions for ARP Publication*. We identify four concrete privacy notions in the context of ARP-data publication. Each notion is defined over a different type of information that needs to be privacy-protected as well as the probability that this protection holds. Specifically, they are derived from the widely-known *differential privacy* [17] notion, which allows us to mathematically prove whether a specific algorithm adheres to any of these notions. We argue that this is a necessary and essential step towards designing, implementing and deploying any privacy-preserving approach into the real world. Without it, it is doubtful whether any meaningful guarantee can be obtained from our approaches.

(ii) *Releasing ARP for Anomaly Detection with Various Degrees of Privacy*. We present four approaches capable of privately releasing ARP data that still preserves the utility of detecting LAN anomalies. Our approach provides a wide range of privacy-preserving degrees, making them suitable to different scenarios:

  (a) The first approach requires small additive perturbations to the input ARP data in exchange for privacy protection of user relationship
  (b) The second approach perturbs the input data by a relatively higher amount but it can attain a stronger privacy protection guarantee for each individual LAN device/user
  (c) The third and fourth are variants of the first two approaches that require even smaller data perturbations; however, they sacrifice some small probability that the privacy guarantee will not hold, making them an appropriate option for scenarios where data utility needs to be maximized

(iii) *Practicality via Real-World Deployment*. We demonstrate practicality of our approaches by implementing and deploying them as part of a large-scale real-world project, called ASEAN-Wide Cyber-Security Research Testbed Project (https://www.nict.go.jp/en/asean_ivo/ASEAN_IVO_2020_Project03.html). Overall, the aim of this project is three-fold: (1) to capture network data from multiple LANs across the ASEAN region, (2) to determine malware behaviors based on the captured data, and (3) to make the captured data sharable in the public domain. Our work fits perfectly in this project as it fulfills the third goal by providing a privacy-preserving mechanism for releasing captured ARP data.

(iv) *Evaluation on Real-World Dataset*. We evaluate our approaches on a real-world ARP dataset captured from 3 LANs over 30 weeks. The experimental result shows feasibility of our approaches as they introduce only low error values (< 10 in the root-mean-square error) to the original data. In addition, we assess utility of the released data by testing it on the existing LAN anomaly detector [7]. The result is promising as our approaches can achieve 75% anomaly detection rate.

Organization: the rest of the paper is organized as follows: Section 2 overviews existing work related to LAN anomaly detection and differential privacy. The background in Address Resolution Protocol and differential privacy are discussed in Section 3. Section 4 describes the system and adversarial models targeted in this work. Section 5 presents privacy notions in the context of releasing ARP data. Sections 6 and 7 present four approaches and prove that they satisfy privacy notions defined in the previous section. Experiments are carried out and reported in Section 8. Several issues are discussed in Section 9. Finally, the paper concludes in Section 10.

2. Related Work

2.1. Differential Privacy in Anomaly Detection. To the best of our knowledge, there has been no prior work that proposes a release mechanism for ARP data with differential privacy guarantees while retaining the utility of anomaly detection in the LAN setting. The closest related work can be found in [18], where the authors employ PINQ differential privacy framework [19] to detect network-wide traffic anomalies. The main difference between our work and the work in [18] lies in the type and magnitude of the released data as well as the privacy guarantee. The work in [18] aims to privately release *link-level traffic volumes of ISP* whose overall value tends to be much larger than noise introduced by any differentially-private release mechanism. On the other hand, our work operates on more restricted input (ARP-degree) which generally contains a much smaller value, making it more noise-sensitive than ISP's traffic volume. Reducing this sensitivity poses a main challenge

addressed in this work. Further, the work in [18] provides *no* privacy protection guarantee for individual network users. Achieving this guarantee is nontrivial, as discussed in Section 6.2.

Besides the work in [18], several existing work focuses on providing anomaly detection with differential privacy guarantees in non-networked settings, e.g., web browsing [20], social network [21], health care [22], or syndrome surveillance [23]. Due to the difference in the target setting, the aforementioned techniques are not directly applicable to our work.

2.2. LAN Anomaly Detection. There are a number of existing research that aims to detect anomalies in LAN *without* providing privacy protection. Zhang et al. [24] present an approach based on honeypot to detect malicious LAN activities. Yeo et al. [25] propose 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 WLAN 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 works focus on detecting LAN anomalies based on ARP-related data. Whyte et al. [26] propose an anomaly detection approach that distinguishes anomalous activities through statistical analyses of ARP traffic. Yasami et al. [8] propose to model normal ARP traffic behaviors using Hidden Markov Model. Farahmand et al. [27] detect LAN anomalies based on four features: traffic rate, burstiness, dark space, and sequential scan. Matsufuji et al. [7] present an anomaly detection algorithm based on the degree of destination of ARP requests.

3. Background

3.1. Address Resolution Protocol (ARP). In a nutshell, ARP is a request-response protocol that provides a mapping between dynamic IP addresses and permanent link-layer addresses (also known as MAC addresses), allowing one computer to discover a MAC address of another from its IP address. This protocol is essential in a LAN environment since it enables communication between any two computers within the same subnetwork as follows:

In LAN, when one computer needs to connect with another, it uses ARP to broadcast a request asking for the MAC address associated with the IP address of the destination computer. Therefore, an ARP request contains the requester's IP and MAC addresses as well as the destination's IP address. Upon receiving the ARP request, every computer checks whether the received IP address matches with one of its network interfaces. If it does, it unicasts an ARP response back to the requester along with its IP and MAC addresses. At the end of this process, the requester successfully retrieves the destination's MAC address and can use this information to construct Ethernet frames for transmitting subsequent data to the target computer.

Similar to other network protocols, ARP involves using sensitive data that has previously been shown to be directly (e.g., IP address) or indirectly (e.g., traffic volume [16]) linkable to the identity of network users. Hence, this privacy concern must be taken into account when designing an approach for releasing ARP data.

3.2. Differential Privacy (DP). Consider a setting in which there are n users who send individual data to a trusted curator. The curator then applies an algorithm $\mathscr{M}$ 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 viewpoint of this notion given in a seminal paper by Dwork, McSherry, Nissim, and Smith [17]. First, we say that two databases X and X' are *neighboring* if they differ by exactly one database entry. The differential privacy is then satisfied if changing X to X' does not change the probability of observing an output of $\mathscr{M}$ by very much. With differential privacy, presence of a single entry will not affect the published output by much. Therefore, outputs from a differentially-private algorithm cannot be used to infer about any single entry from the input dataset.

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

$$\mathbb{P}(\mathscr{M}(X) \in S) \leq e^{\varepsilon}\mathbb{P}(\mathscr{M}(X') \in S) + \delta, \tag{1}$$

where ε is referred as a privacy budget. We will refer to $(\varepsilon, 0)$-DP as ε-DP. Intuitively, smaller values of ε and δ lead to a stronger privacy guarantee. Conversely, higher values of ε and δ imply a weaker guarantee with possibly better utility/accuracy of the released data.

A related notion of differential privacy is the concentrated differential privacy, which aims to control the moments of the *privacy loss variable*: $f(Y) = \mathbb{P}(\mathscr{M}(X) = Y)/\mathbb{P}(\mathscr{M}(X') = Y)$, where Y is distributed as $\mathscr{M}(X)$.

Definition 2 (Rényi divergence). Let P and P' be probability densities. The Rényi divergence of order $\lambda \in (1, \infty)$ between P and P' is defined as

$$\begin{aligned} D_{\lambda}(P \| P') &= \frac{1}{\lambda - 1}\log \int P(y)^{\lambda} P'(y)^{1-\lambda} \mathrm{d}y \\ &= \frac{1}{\lambda - 1}\log \mathbb{E}_{y \sim P}\left[\frac{P(y)^{\lambda-1}}{P'(y)^{\lambda-1}}\right]. \end{aligned} \tag{2}$$

Definition 3 (concentrated differential privacy [28]). An algorithm $\mathscr{M}: \mathscr{X} \longrightarrow \mathscr{Y}$ satisfies ρ-*zero-concentrated differential privacy* (ρ-zCDP) if, for every pair of neighboring datasets X and X' and every$\lambda \in (1, \infty)$,

$$D_{\lambda}(\mathscr{M}(X) \| \mathscr{M}(X')) \leq \lambda\rho. \tag{3}$$

One useful property of the differential privacy is that it is preserved under post-processing.

Proposition 1 (postprocessing [29]). *For any (ε, δ)-DP (ρ-zCDP) algorithm $\mathcal{M}: \mathcal{X} \longrightarrow \mathcal{Y}$ and arbitrary random function $f: \mathcal{Y} \longrightarrow \mathcal{Z}$, the algorithm $f \circ \mathcal{M}$ is also (ε, δ)-DP (ρ-zCDP).*

There may be some certain situations in which we want to apply multiple DP algorithms, e.g., releasing continual or time-series data. In this case, the resulting algorithm is also differentially private. However, every new DP algorithm comes with a cost of privacy loss, as stated in the following proposition.

Proposition 2 (composition [29]). *For any (ε, δ)-DP (ρ-zCDP) algorithms $\mathcal{A}_i: \mathcal{X} \longrightarrow \mathcal{Y}_i$ for $i \in [k]$, the algorithm $\mathcal{A}_{[k]}: \mathcal{X} \longrightarrow \prod_{i=1}^{k} \mathcal{Y}_k$ defined by $\mathcal{A}_{[k]}(X) = (\mathcal{A}_1(X), \ldots, \mathcal{A}_k(X))$ is $(k\varepsilon, k\delta)$ -DP ($k\rho$-zCDP).*

To introduce one of the most ubiquitous ε-DP algorithms, we start with the ℓ_1-*sensitivity* of a randomized algorithm $\mathcal{M}: \mathcal{X} \longrightarrow \mathbb{R}^k$, 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:

$$\Delta^{\mathcal{M}} = \max_{\text{neighbor } X, X'} \left\| \mathcal{M}(X) - \mathcal{M}(X') \right\|_1. \tag{4}$$

Theorem 1 (Laplace mechanism [29]). *Let $\mathcal{M}: \mathcal{X} \longrightarrow \mathbb{R}^k$ be an algorithm with sensitivity $\Delta^{\mathcal{M}}$ and Y_i be a noise generated by sampling from a Laplace distribution at scale $= \Delta^{\mathcal{M}}/\varepsilon$, i.e., $Y_i \sim Laplace(\Delta^{\mathcal{M}}/\varepsilon)$, then the randomized algorithm $\mathcal{A}$ defined by*

$$\mathcal{A}(X) = \mathcal{M}(X) + (Y_1, \ldots, Y_k), \tag{5}$$

is ε-DP.

In addition to the Laplace mechanism, the Gaussian mechanism is also commonly used to provide ρ-zCDP:

Theorem 2 (Gaussian mechanism [28]). *Let $\mathcal{M}: \mathcal{X} \longrightarrow \mathbb{R}^k$ be an algorithm with sensitivity $\Delta^{\mathcal{M}}$ and Y_i be a noise generated by sampling from a Gaussian distribution at scale $\Delta^{\mathcal{M}}/\sqrt{2\rho}$, i.e., $Y_i \sim N(0, (\Delta^{\mathcal{M}})^2/2\rho)$, then the randomized algorithm $\mathcal{A}$ defined by*

$$\mathcal{A}(X) = \mathcal{M}(X) + (Y_1, \ldots, Y_k), \tag{6}$$

is ρ-zCDP.

In view of Proposition 2, a composition of N Laplace mechanisms at scale $N\Delta^{\mathcal{M}}/\varepsilon = O(N)$ is ε -DP, while that of N Gaussian mechanisms at scale $\Delta^{\mathcal{M}}\sqrt{N/2\rho} = O(\sqrt{N})$ is ρ-zCDP. We see that, for successive use of a DP mechanism, the Gaussian mechanism gives comparatively smaller noise than the Laplace mechanism. The following lemma shows how the two definitions of differential privacy are related.

Lemma 1 (see [28]). *Any ρ-zCDP algorithm is also an (ε, δ)-DP algorithm for any given $\delta \rangle 0$ and*

$$\varepsilon = \rho + 2\sqrt{\rho \log\left(\frac{1}{\delta}\right)}. \tag{7}$$

Conversely, for any given ε and $\delta \rangle 0$, any ρ-zCDP algorithm where

$$\rho = \left(\sqrt{\log\left(\frac{1}{\delta}\right) + \varepsilon} - \sqrt{\log\left(\frac{1}{\delta}\right)} \right)^2, \tag{8}$$

is also an (ε, δ)-DP algorithm.

4. System and Adversarial Models

Figure 1 illustrates the system model considered in this work. We consider a system in which an entity, called Admin, possesses a LAN consisting of nUsers (i.e., computing devices). In addition, Admin introduces a monitoring device to this LAN in order to observe ARP requests of all Users. 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," since this time scale allows us to use data collected from a long period of time without losing too much privacy budget from the composition (Proposition 2). V_j is denoted the result after appending all ARP requests of all User-s generated in week j, i.e. $V_j = \{V_{j1}, V_{j2}, \ldots, V_{jn}\}$.

As shown in Figure 1, our system starts by having the monitoring node (periodically) send aggregate ARP requests—$V = \{V_1, \ldots, V\}_t$—to Admin, corresponding to step ❶ in Figure 1. Admin is interested in learning whether the LAN *as a whole* has had any anomalous activities for the last t weeks *in a private way*. Thus, in step ❷, he proceeds to apply a certain algorithm Algo with the goal of hiding sensitive information from the input V and then releases the output D to an external entity Analyst in step ❸. In step ❹, 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. We summarize notation used throughout the paper in Table 1

4.1. Adversarial 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 attempt to learn sensitive information about Users or their relationship, and use it for his own benefits.

4.2. Goal and Scope. In this work, we focus on addressing privacy concerns in the aforementioned system, where data from LAN is exposed to an external party. Hence, we do not consider other LAN settings capable of handling and processing this data locally, e.g., LANs in a large corporate with its own internal anomaly detection tool.

The goal of this work is to design approaches that can be appropriately used as the algorithm Algo in step ❷ of Figure 1. In other words, our approaches must allow the

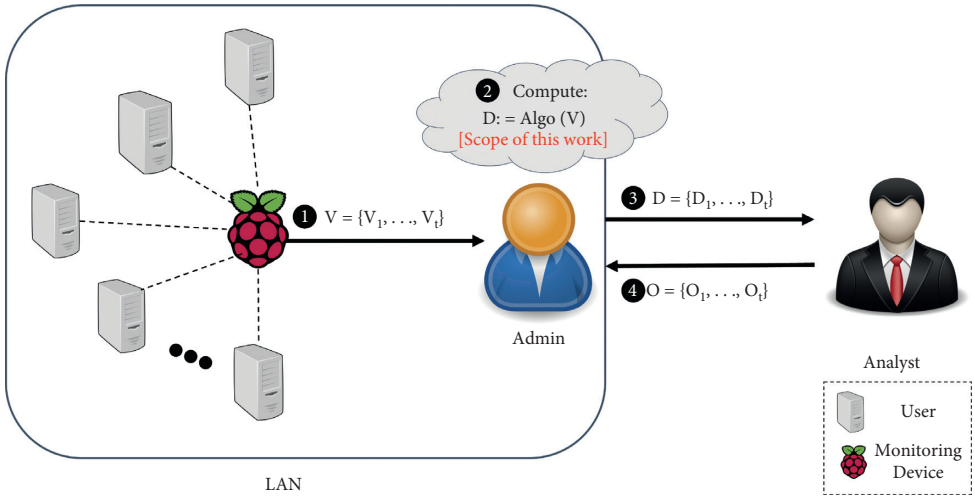

Figure 1: System model considered in this work.

Table 1: Notation.

Differential Privacy (DP) Notation	
ε	Privacy budget
δ	Probability of failing DP guarantees
$\Delta^{\mathcal{M}}$	Sensitivity of algorithm $\mathcal{M}$
Laplace (b)	Laplace distribution with mean 0 and scale b
$N(\mu, \sigma^2)$	Normal distribution with mean μ and standard deviation σ
System notation	
n	Number of LAN user s
t	Number of data collection intervals
V_{jk}	User k's ARP requests aggregate at interval j
$V_j = \{V_{j1}, \ldots, V_{jn}\}$	Aggregate ARP requests of all LAN user s at interval j
$V = \{V_1, \ldots, V_t\}$	Aggregate ARP requests of all LAN user s from interval 1 to t
$D = \{D_1, \ldots, D_t\}$	Output after applying privacy-preserving algorithm
$O = \{O_1, \ldots, O_t\}$	Anomaly detection output

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.

5. DP Notions for ARP-Request Data

In this section, we describe 4 variants of differential privacy notions related to our system model. The summary of DP notions discussed throughout this Section is shown in Table 2.

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 Users producing 4 ARP requests over a specific time interval. We define the (ARP-request) "degree" of User k as the number of Users that receives ARP requests from User k. In this example, the degrees of User 1, 2, and 3 are 2, 2, and 0, respectively.

Table 2: Summary of DP notions for ARP-request data.

Notion	Definition #	Protected info.	Protection prob.
(ε, δ)-Edge-DP	4	ARP requests	1-δ
ε-Edge-DP	5	ARP requests	1
(ε, δ)-Node-DP	6	Users	1-δ
ε-Node-DP	7	Users	1

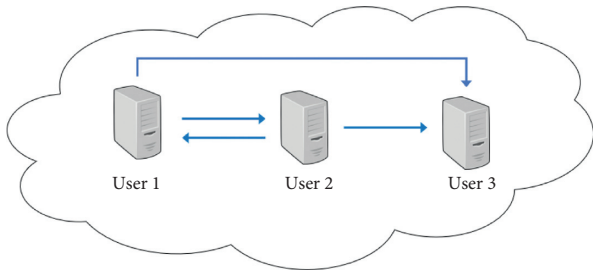

Figure 2: Illustration of a LAN with 3 User-s and 4 ARP requests (represented by arrows).

Using this model, 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 in the same time interval. The degree of User k is then equivalent to the number of directed edges originating from User k.

As a directed graph, V_j can not directly 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 was prior work focusing on expressing differential privacy of a graph database. Specifically, the work in [30] 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. Likewise, they are *node-neighboring* if they differ by a single node.

We now proceed to present two notions of privacy in edge-neighboring graphs:

Definition 4 ((ε, δ)-edge-DP). Let $\mathcal{G}$ be the set of graphs between Users. An algorithm $\mathcal{M}: \mathcal{G} \longrightarrow \mathcal{Y}$ satisfies (ε, δ)-edge-differential privacy or (ε, δ)-edge-DP if, for every pair of edge-neighboring graphs G and G' and every subset $S \subseteq \mathcal{Y}$,

$$\mathbb{P}(\mathcal{M}(G) \in S) \leq e^{\varepsilon}\mathbb{P}(\mathcal{M}(G') \in S) + \delta. \tag{9}$$

Definition 5 (ε-edge-DP). An algorithm satisfies ε-edge-differential privacy (ε-edge-DP) if and only if it satisfies $(\varepsilon, 0)$-edge-DP.

Since an edge in our system refers to ARP requests between a pair of Users, Definitions 4 and 5 provide privacy protection for these ARP requests. This means that an algorithm satisfying ε-edge-DP/ (ε, δ)-edge-DP is guaranteed to reveal no information about all ARP requests exchanged between any pair of Users, *resulting in hiding the ARP relationship of all Users*. This, for example, could hide the source of infection in LAN as it is common for malware to utilize ARP as the first step to discover and infect other LAN User-s.

Nonetheless, the guarantee provided by these definitions is not strong enough to protect privacy of individual Users. To achieve this stronger guarantee, we adopt the following notions:

Definition 6 ((ε, δ)-node-DP). Let $\mathcal{G}$ be the set of graphs between Users. An algorithm $\mathcal{M}: \mathcal{G} \longrightarrow \mathcal{Y}$ satisfies (ε, δ)-node-differential privacy or (ε, δ)-node-DP if, for every pair of node-neighboring graphs G and G' and every subset $S \subseteq \mathcal{Y}$,

$$\mathbb{P}(\mathcal{M}(G) \in S) \leq e^{\varepsilon}\mathbb{P}(\mathcal{M}(G') \in S) + \delta. \tag{10}$$

Definition 7 (ε-node-DP). An algorithm satisfies ε-node-differential privacy (ε-node-DP) if and only if it satisfies $(\varepsilon, 0)$-node-DP.

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/ (ε, δ)-node-DP prevents information leakage about presence or absence of any individual User.

Remark 1. Recall δ represents an upper bound of the probability that an algorithm fails to satisfy the ε-DP notion. As an example, an algorithm satisfying (ε, δ)-node-DP has at most δ probability that will leak some information about an individual node in a graph. To make (ε, δ)-edge/node-DP notions meaningful in practice, one must minimize this failure probability by ensuring that δ is negligible in terms of number of data points ($\#p$) considered in the DP notion [29]. One way to achieve this is to set δ to: $\delta = \delta'/\#p$ for some small δ'.

In (ε, δ)-node-DP notion, $\#p$ is the number of nodes; whereas, in (ε, δ)-edge-DP, $\#p$ corresponds to the number of possible directed edges $\approx (\#\text{nodes})^2$. Thus, it is easy to see that δ in (ε, δ)-edge-DP must be set smaller than that in (ε, δ)-node-DP in order to attain the negligible probability.

6. Releasing ARP-Request Data with ε-Edge/ Node-DP

In this section, we present two approaches, called *naïve* and *histogram-based*; the former guarantees ε-edge-DP while the latter is proven to satisfy the ε-node-DP notion. Later in Section 7, we describe variants of these approaches that satisfy the more relaxed (ε, δ)-edge/node-DP notions.

6.1. Naïve Approach. The naïve approach is described 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.

Theorem 3. *The naïve approach as described in Algorithm 1 is ε-edge-DP.*

Proof. 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, $\Delta^{\mathcal{M}} = |D_j - D_j'| \leq 1$ and we have the following Laplace mechanism $\mathcal{A}'$ (line 2-3) guarantee the ε/t-node DP:

$$\mathcal{A}'(V_j) = \mathcal{M}(V_j) + Y_j, \tag{11}$$

where $Y_j \sim \text{Laplace}(t/\varepsilon)$ (line 3).

Algorithm 1 can then be represented as

$$\mathcal{A}(V) = \text{P}(\mathcal{A}'(V_1), \ldots, \mathcal{A}'(V_t)), \tag{12}$$

where P is a postprocessing function (line 4-5) that: (i) precludes a negative output by thresholding it to 0, and (ii) rounds a nonnegative privatized value into the closest integer in order to prevent the floating point attack [31].

Input: $V = \{V_1, V_2, \ldots, V_t\}$,
t,
ε
Output: $D = \{D_1, D_2, \ldots, D_t\}$
(1) **for** $j = 1$ **to** t **do**
(2) $D_j \longleftarrow \text{Sum}(\text{Degree}(V_j))$
(3) $D_j \longleftarrow D_j + \text{Laplace}(t/\varepsilon)$
(4) **if** $D_j > 0$ **then** $D_j \leftarrow \text{int}(D_j)$
(5) **else** $D_j \longleftarrow 0$
(6) **end**

Algorithm 1: Naïve Approach.

By Proposition 1 and 2, we can conclude that this algorithm is $t\varepsilon/t$-edge-DP or ε-edge-DP.

To prevent excessive information loss, one needs the Laplace noise to be smaller than D_j, i.e., $t/\varepsilon \langle \mathbb{E}[D_j]$ or $\varepsilon \rangle t/\mathbb{E}[D_j]$. This can be achieved in realistic settings, e.g., $\varepsilon = 2$ in our experiment (Section 8) 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 Users, it is no longer feasible to preserve both privacy and utility at the same time.

6.2. Histogram-Based Approach. As seen in the previous subsection, the naïve approach cannot be used to satisfy ε-node-DP in practice due to its high sensitivity, leading to too strong additive noises which in turn significantly lower 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 histogram-based approach is shown in Algorithm 2. The rationale behind this approach is to transform the degree data in such a way that its 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 disjoint 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), threshold and round the resulting value to the closest integer (line 5-6) and finally return this noisy histogram as an output.

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

Theorem 4. *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 $\mathcal{M}: \mathcal{G} \longrightarrow \mathbb{R}^k$ 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, $\Delta^{\mathcal{M}} = |\mathcal{M}(V) - \mathcal{M}(V_j')| \leq 1$.

Observe that line 2-7 of Algorithm 2 can be written as a randomized algorithm $\mathcal{A}': \mathcal{G} \longrightarrow \mathbb{R}^k$ defined by

$$\mathcal{A}'\left(V_j\right) = \mathrm{P}\left(\mathcal{M}\left(V_j\right) + \left(Y_1, \ldots, Y_k\right)\right), \tag{13}$$

where $Y_i \sim \text{Laplace}(t/\varepsilon)$ and P corresponds to the *threshold-then-round* function computed on all bin counts (line 5-6). It follows from Theorem 1 and Proposition 1 that $\mathcal{A}'$ is ε/t-node-DP.

Then, we can define Algorithm 2 as a randomized algorithm $\mathcal{A}$ as follows:

$$\mathcal{A}(V) = \left(\mathcal{A}'\left(V_1\right), \ldots, \mathcal{A}'\left(V_t\right)\right). \tag{14}$$

By Proposition 2, we have that the histogram-based approach (described in Algorithm 2) is $t\varepsilon/t$-node-DP or ε-node-DP. □

7. Releasing ARP-Request Data with (ε, δ)-Edge/Node-DP

The approaches in the previous section require adding a noise proportional to t, which may not scale well in practice when t is large. We explore an alternative by instead adopting the Gaussian Mechanism in order to reduce additive noise from $O(t)$ to $O(\sqrt{t})$. We call these variants, *naïve-δ* and *histogram-based-δ*, which guarantee (ε, δ)-edge-DP and (ε, δ)-node-DP, respectively.

7.1. Naïve-δ Approach. In conjunction with the naïve approach (Algorithm 1) which gives a strong privacy guarantee by adding considerably large amount of noises, we develop here another approach that adds less noises, but provides a weaker (ε, δ)-edge DP guarantee. The algorithm is described

Input: $V = \{V_1, V_2, \ldots, V\}_t$, t, ε
Output: $D = \{D_1, D_2, \ldots, D_t\}$
(1) **for** $j = 1$ **to** t **do**
(2) $D_j \longleftarrow$ Histogram (Degree (V_j))
(3) **foreach** bin $\in D_j$ **do**
(4) bin · count $\longleftarrow$ bin · count + Laplace (t/ε)
(5) **if** bin · count > 0 **then**
bin · count $\longleftarrow$ int (bin · count)
(6) **else** bin · count $\longleftarrow 0$
(7) **end**
(8) **end**

Algorithm 2: Histogram-based Approach.

in Algorithm 3. Similar to Algorithm 1, we round the noisy outputs to the nearest integers to protect the data from floating point attacks. In the rest of this section, we discuss nontrivial details of this approach and show that it indeed satisfies (ε, δ)-edge DP.

Theorem 5. *The naïve-δ approach as described in Algorithm 3 is (ε, δ)-edge-DP.*

Proof. 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 3 of Algorithm 3). As in the proof of Theorem 3, the edge-sensitivity $\Delta^{\mathcal{M}}$ satisfies $\Delta^{\mathcal{M}} \leq 1$. Observe that line 3-6 of Algorithm 3 can be written as a randomized algorithm $\mathcal{A}' : \mathcal{G} \longrightarrow \mathbb{R}^k$ defined by

$$\mathcal{A}'\left(V_j\right) = \mathrm{P}\left(\mathcal{M}\left(V_j\right) + \left(Y_1, \ldots, Y_k\right)\right), \tag{15}$$

where $Y_i \sim N(0, t/2\rho)$ and P corresponds to the *threshold-then-round* function computed on all bin counts (line 5-6). It follows from Theorem 2 and Proposition 1 that $\mathcal{A}'$ is ρ/t-zCDP.

Then, we can define Algorithm 3 as a randomized algorithm $\mathcal{A}$ as follows:

$$\mathcal{A}(V) = \left(\mathcal{A}'(V_1), \ldots, \mathcal{A}'(V_t)\right). \tag{16}$$

By Proposition 2, we have that the Algorithm 3 is $t\rho/t$-zCDP or ρ-zCDP. Using Lemma 1 and recalling the definition of ρ in line 1 of Algorithm 3, we conclude that this algorithm is also (ε, δ)-edge-DP. □

7.2. Histogram-Based-δ Approach. We aim to construct an (ε, δ)-node-DP with less noises compared to the ε-node-DP algorithm in Section 6.2. We still rely on a histogram-based approach as it has small sensitivity upon adding/removing a node. Our histogram-based-δ approach is described in Algorithm 4.

Theorem 6. *The histogram-based-δ approach as described in Algorithm 4 is (ε, δ)-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 $\mathcal{M} : \mathcal{G} \longrightarrow \mathbb{R}^k$ 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. As in the proof of Theorem 4, the node-sensitivity $\Delta^{\mathcal{M}}$ satisfies $\Delta^{\mathcal{M}} \leq 1$

Looking at Algorithm 4, we observe that line 3-7 can be written as a randomized algorithm $\mathcal{A}' : \mathcal{G} \longrightarrow \mathbb{R}^k$ defined by

$$\mathcal{A}'\left(V_j\right) = \mathrm{P}\left(\mathcal{M}\left(V_j\right) + \left(Y_1, \ldots, Y_k\right)\right), \tag{17}$$

where $Y_i \sim N(0, t/2\rho)$ and P corresponds to the *threshold-then-round* function computed on all bin counts (line 6-7). It follows from Theorem 2 and Proposition 1 that $\mathcal{A}'$ is ρ/t-node-DP.

Then, we can define Algorithm 4 as a randomized algorithm $\mathcal{A}$ as follows:

$$\mathcal{A}(V) = \left(\mathcal{A}'(V_1), \ldots, \mathcal{A}'(V_t)\right). \tag{18}$$

By Proposition 2, we have that the histogram-based approach (described as in Algorithm 4) is $t\rho/t$-zCDP or ρ-zCDP. From the definition of ρ in line 1 of Algorithm 4, we conclude using Lemma 1 that this algorithm is also (ε, δ)-node-DP. □

8. Evaluation

In this section, we evaluate our approaches by deploying them as part of a large-scale research project and reporting their utility from a real-world dataset extracted from such project.

8.1. Real-World Deployment

8.1.1. Background. ASEAN-Wide Cyber-Security Research Testbed Project is a large-scale research project with collaboration between multiple universities primarily located in Southeast Asia including Prince of Songkla University, Thailand (PSU), Universitas Brawijaya, Indonesia (UB), University of Computer Studies Yangon, Myanmar (UCSY), Institute of Technology of Cambodia, Cambodia (ITC), University of Information Technology, Myanmar (UIT), and The University of Tokyo, Japan (UT). The ultimate goal of

Input: $V = \{V_1, V_2, \ldots, V_t\}$, t, ε, δ
Output: $D = \{D_1, D_2, \ldots, D_t\}$
(1) $\rho \longleftarrow (\sqrt{\log(1/\delta) + \varepsilon} - \sqrt{\log(1/\delta)})^2$
(2) **for** $j = 1$ **to** t **do**
(3) $\quad D_j \longleftarrow \text{Sum}(\text{Degree}(V_j))$
(4) $\quad D_j \longleftarrow D_j + N(0, t/2\rho)$
(5) $\quad$ **if** $D_j > 0$ **then** $D_j \leftarrow \text{int}(D_j)$
(6) $\quad$ **else** $D_j \longleftarrow 0$
(7) **end**

Algorithm 3: Naïve-δ Approach.

Input: $V = \{V_1, V_2, \ldots, V_t\}$, t, ε, δ
Output: $D = \{D_1, D_2, \ldots, D_t\}$
(1) $\rho \longleftarrow (\sqrt{\log(1/\delta) + \varepsilon} - \sqrt{\log(1/\delta)})^2$
(2) **for** $j = 1$ **to** t **do**
(3) $\quad D_j \longleftarrow \text{Histogram}(\text{Degree}(V_j))$
(4) $\quad$ **foreach** bin $\in D_j$ **do**
(5) $\quad\quad$ bin $\cdot$ count $\longleftarrow$ bin $\cdot$ count $+ N(0, t/2\rho)$
(6) $\quad\quad$ **if** bin $\cdot$ count$\rangle$0 **then**
$\quad\quad$ bin $\cdot$ count $\longleftarrow$ int(bin $\cdot$ count)
(7) $\quad\quad$ **else** bin $\cdot$ count $\longleftarrow 0$
(8) $\quad$ **end**
(9) **end**

Algorithm 4: Histogram-based-δ Approach.

this project is to create a real-world public testbed of malware behaviors captured in ASEAN countries.

Independent of our work, the first phase of this project involves capturing, collecting and analyzing LAN data in Southeast Asian countries. To achieve this task, a small monitoring device, implemented atop of a raspberry-Pi 3B in Figure 3, is introduced and placed into several LANs across the ASEAN region. This monitoring device observes and captures the network traffic flowing within a LAN and periodically outputs the captured data to our server, in which such data is analyzed and a model of ASEAN malware is eventually created.

8.1.2. Deployment. Our work plays an important role in the second phase of this research project. It allows us to privately share aggregate ARP data collected from the previous phase with other project members as well as to the public domain.

Our approaches enable a release mechanism of ARP-request data that still retains the utility of LAN anomaly detection. To assess utility, we evaluated our approaches on a subset of data captured and extracted from this research project.

The extracted dataset contains all ARP-request data observed and collected from 3 real-world LANs over a 30-week period. These LANs are located in: (1) The University of Tokyo, Japan (thus, its dataset is labeled as JPN), (2) Prince of Songkla University–Phuket Campus, Thailand (HKT) and (3) Prince of Songkla University–Hatyai Campus, Thailand (HDY). Details about these monitored LANs can be found in Table 3.

8.1.3. Parameter Selection. As we collected ARP requests over a 30-week period, $t = 30$. The naïve approach involves no other parameters. Meanwhile, the histogram-based approach consists of an additional set of 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 noise injected by our approach would become too large, severely decreasing utility of the released data. To avoid this problem, we select the number of histogram bins to be relatively small – 3. Specifically, we choose the first two bins to correspond to the number of Users whose degrees are 1 and 2, respectively; the third bin contains the number of User-s with degree ≥3.

Finally, the approaches in Section 7 consist of another parameter δ. Recall from the Remark 1 in Section 5 that δ must be negligible with respect to the number of data points ($\#p$). In other words:

$$\delta = \delta'/\#p \text{ for some small } \delta', \tag{19}$$

In our target system, $\#p$ corresponds to n and n^2 for the node-DP and edge-DP notions, respectively; See Table 3 for the number of Users (n) in each monitored LAN. Unless stated otherwise, we use $\delta' = 0.01$ for all experiments.

Figure 3: Monitoring device (raspberry-Pi 3B) deployed to a LAN.

Table 3: Details of monitored LANs.

Label	Location of LAN			Collection period			Users (n)
	University	City	Country	Start date	End date	# Weeks (t)	
JPN	UT	Tokyo	Japan	Aug 9, 2019	Mar 6, 2020	30	95
HKT	PSU	Phuket	Thailand	Nov 6, 2020	June 4, 2021	30	63
HDY	PSU	Hat Yai	Thailand	Oct 21, 2020	May 19, 2021	30	206

Nonetheless, the impact of different δ' values on the utility is also assessed in the next subsection.

8.2. Utility Assessment: RMSE

8.2.1. RMSE. In the context of differential privacy, one common utility metric is defined as an error between the released privatized values z^* and the nonprivatized aggregates z. We adopt a similar approach and select the root-mean-square error (RMSE) as our first evaluation metric:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(z^*[i] - z[i]\right)^2}. \tag{20}$$

where $z[i]$ and $z^*[i]$ represent the i^{th} data point in z and z^*, respectively. For the naïve approach and its variant, $z[i]$ corresponds to the sum of all User's ARP degrees observed in week i, while $z^*[i]$ refers to the privatized output on the same ARP data. On the other hand, $z[i]$ represents a histogram bin in the histogram-based and histogram-based-δ approaches.

8.2.2. Impact of ε. Recall that ε refers to a privacy budget in the DP notion and a lower value of ε implies stronger privacy, while possibly sacrificing utility.

Figure 4 shows the impact of ε on the utility of the proposed approaches. Unsurprisingly, we achieve lower errors and thus better utility from a higher ε. For all 3 monitored LANs, $\varepsilon = 5$ seems to be a pragmatic choice in order to maintain a low error (< 10) for all approaches.

Next, we show how much utility can be improved by using the approaches in Section 7 instead of their counterparts in Section 6. The result, illustrated in Figure 5, suggests that both naïve-δ and histogram-based-δ approaches enjoy higher utility (i.e., a utility gain) when $\varepsilon \leq 4$. However, as the ε gets larger, this utility gain becomes smaller; in fact, the naïve-δ approach incurs a utility loss when $\varepsilon \geq 8$ for all monitored LANs. This result suggests using the approaches in Section 7 only when one needs stronger privacy, i.e., small ε.

Figure 5 also indicates the histogram-based-δ approach significantly outperforms the naïve-δ approach in terms of the utility gain. For $\varepsilon \leq 4$, the histogram-based-δ approach provides $\geq 28\%$ utility gain, while a smaller amount of utility gain ($\leq 20\%$) can be realized in the naïve-δ approach. This is expected because the histogram-based-δ approach introduces a smaller value of δ (see the Remark 1 in Section 5), making the additive noise smaller and thus resulting in the higher utility gain.

In addition, n also has a direct impact to δ and hence to the overall utility. As seen in Figure 5, among all monitored LANs, HDY has the highest number of Users and therefore suffers the lowest utility gain.

8.2.3. Impact of δ'. We now assess the impact of δ' on the utility of our approaches. Figure 6 shows RMSE of the naïve-δ and histogram-based-δ approaches for different values of δ'. As expected, increasing δ' results in a decrease in RMSE and thus improves the utility of our approaches. This decrease is logarithmic as a function of δ'.

The utility gain of the naïve-δ and histogram-based-δ approaches with respect to their original counterparts is illustrated in Figure 7. Our approaches benefit from the higher utility gain when δ' is larger. For most δ' values, the histogram-based-δ approach provides a positive utility gain over the histogram-based approach. Meanwhile, a utility gain can be achieved from the naïve-δ approach when $\delta' \geq 10^{-3}$.

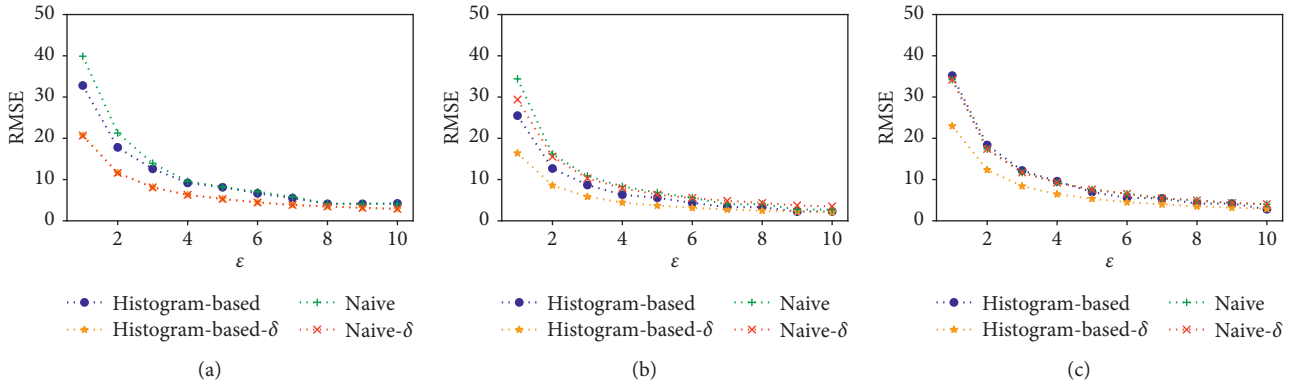

Figure 4: RMSE with different ε values.

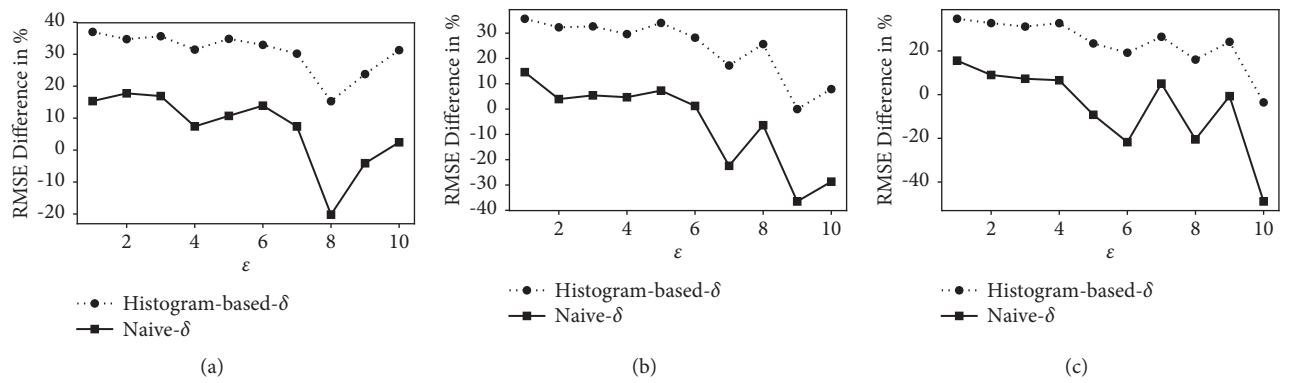

Figure 5: Utility gain (in %) with respect to their ε-edge/node-DP counterparts.

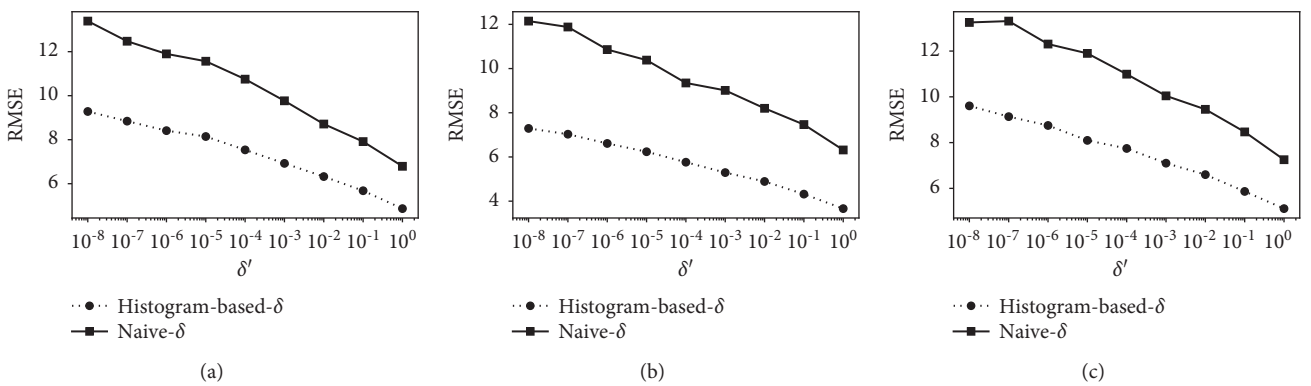

Figure 6: RMSE with different δ' values where $\delta = \delta'/(\#p)$ and ε is fixed to 1.

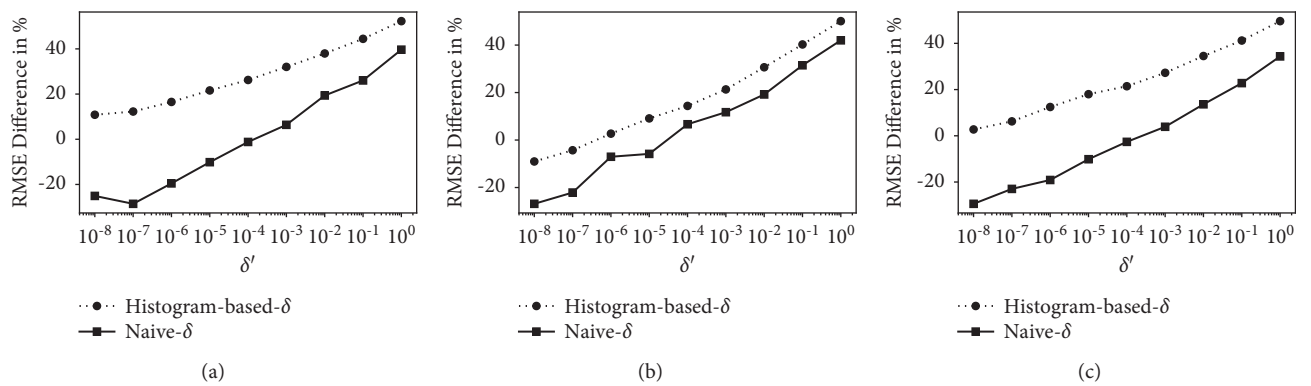

Figure 7: Utility gain (in %) with respect to their ε-edge/node-DP counterparts.

This experimental result suggests that both naïve-δ and histogram-based-δ approaches still provide a utility advantage over their original counterparts even for δ' smaller than 10^{-2} (up to 10^{-3} for the naïve-δ approach and 10^{-6} for the histogram-based-δ approach). In practice, one may choose to opt for smaller δ' if a stronger privacy guarantee is needed.

8.3. Utility Assessment: Anomaly Detection Accuracy

8.3.1. Anomaly Detection Algorithm. In addition to low errors, it is also essential that outputs produced by our approaches can still be useful in identifying anomalous activities in LAN. Hence, we further evaluate utility of our approaches by assessing them via a LAN anomaly detector. In this experiment, we consider our approaches to preserve the utility of anomaly detection if the anomaly detector classifies the privatized data the same way as the original (nonprivatized) data.

For the anomaly detector, we choose an approach based on exponentially weighted moving average and variance [32] proposed by Matsufuji et al. [7] since it is tailored specifically for detecting LAN anomalies based on ARP data, which is also the focus in this work. All parameter values are selected based on the recommendation from [7].

It is worth noting that the anomaly detector in [7] only supports input of type univariate time series. However, the histogram-based approach and its variant produce a multivariate time series output (i.e., a time series of histograms), and hence cannot be used directly as input to the anomaly detector. To address this issue, we perform a simple transformation that converts two consecutive histograms into a single variable using the L_1 distance function; the result of this transformation is then given as input to the anomaly detector. More formally, the transformation is defined as

$$z^*[i] = \|\text{hist}[i] - \text{hist}[i+1]\|_1 \text{ for } i \in \{1, \ldots, t-1\}. \quad (21)$$

8.3.2. Metrics. In this experiment, we evaluate utility of our approaches using two metrics: true positive rate (TPR) and F_1 score. In particular, we consider $z^*[i]$, a noisy data point produced by our approach, to be a true positive (TP) if the anomaly detector classifies both $z^*[i]$ and $z[i]$ as an anomaly, where $z[i]$ represents the original nonprivatized counterpart. $z^*[i]$ is a false positive (FP) if the anomaly detector finds an anomaly in $z^*[i]$ but not in $z[i]$. A true negative (TN) and a false negative (FN) are also defined similarly.

Based on these definitions, TPR and F_1 metrics can be formulated as

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \qquad F_1 = \frac{\text{TP}}{\text{TP} + 0.5\,(\text{FP} + \text{FN})}. \quad (22)$$

A high value of TPR implies that a high percentage of anomalies detected in the original data is also captured as an anomaly in the privatized data. On the other hand, a high value of F_1 implies relatively small values of FP and FN compared to TP.

8.3.3. Results. Figures 8 and 9 show the utility of our approaches evaluated using TPR and F_1 metrics, respectively. First, we can see that ε does not affect utility of the naïve and naïve-δ approaches as both approaches still provide almost perfect utility scores in all monitored LANs.

On the other hand, the histogram-based and histogram-based-δ approaches yield low utility for small ε. The utility scores then become higher as ε increases. For HKT, both approaches achieve a reasonable score of > 0.75 with $\varepsilon = 5$. Meanwhile, ε must be set to 6 in order to achieve the same utility score in HDY. JPN requires the highest $\varepsilon (= 12)$ in order for the histogram-based-δ approach to perform 75% TPR.

Lastly, the results also confirm that the histogram-based-δ approach significantly outperforms the naïve-δ approach in terms of utility. Thus, we recommend to deploy the histogram-based-δ approach over the histogram-based approach when one needs to publish ARP-request data with user privacy protection (i.e., corresponding to the node-DP notion); whereas, if edge-DP is sufficient, the naïve approach is a more reasonable choice over the naïve-δ approach as the former provides a stronger privacy guarantee while both approaches achieve the similar utility performance.

8.3.4. Comparison with RMSE. In most cases, the utility results from TPR and F_1 metrics are consistent with the previous results measured using RMSE in Section 8.2. That is, a higher ε leads to higher utility with lower RMSE and higher TPR and F_1. On the other hand, an extremely low value of ε (e.g., $\varepsilon = 1$) renders the output data useless as it can no longer be used to reveal anomalies due to its low TPR/F_1. There is, however, one exception: the naïve and naïve-δ approaches surprisingly can still attain high TPR and F_1 utility despite low ε. This indicates that such approaches are more robust to additive noises than other approaches.

9. Discussion

9.1. ARP Fields. Our approaches take as input ARP-degree data, which in turn makes use of only 5 fields in ARP packets: SHA, SPA, THA, TPA, and OPER. In this work, we choose to discard the rest of the ARP fields (i.e., Hardware Type/Length (HTYPE/HLEN) and Protocol Type/Length (PTYPE/PLEN)) from our analysis. This is because, in practice, these discarded fields usually have fixed values that contain neither sensitive information nor anything meaningful to our approaches. For instance, since ARP is only applicable to IPv4, the PLEN field is always set to the value of 4 indicating the size of an IPv4 address; or HTYPE usually contains the value of 1 representing the ubiquitous Ethernet hardware type. As these fields are generally constant for all

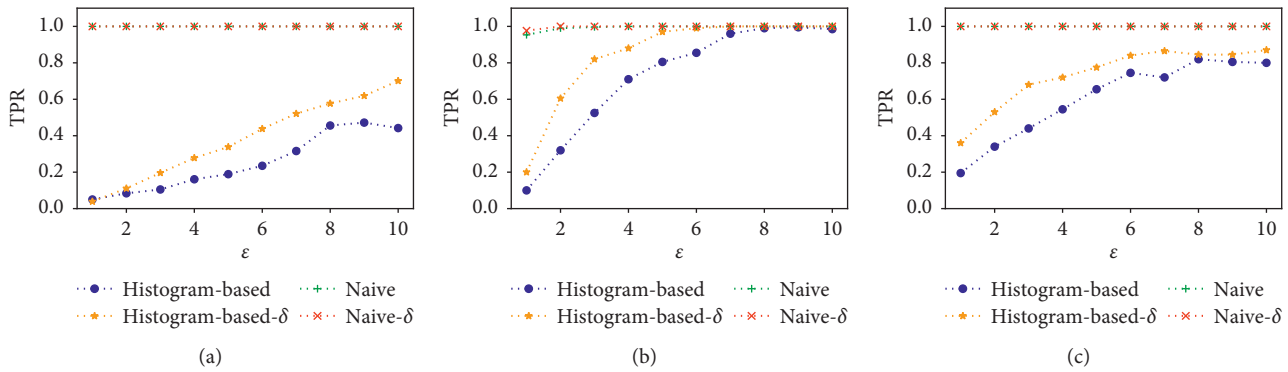

Figure 8: TPR result for different ε in all 3 monitored LANs.

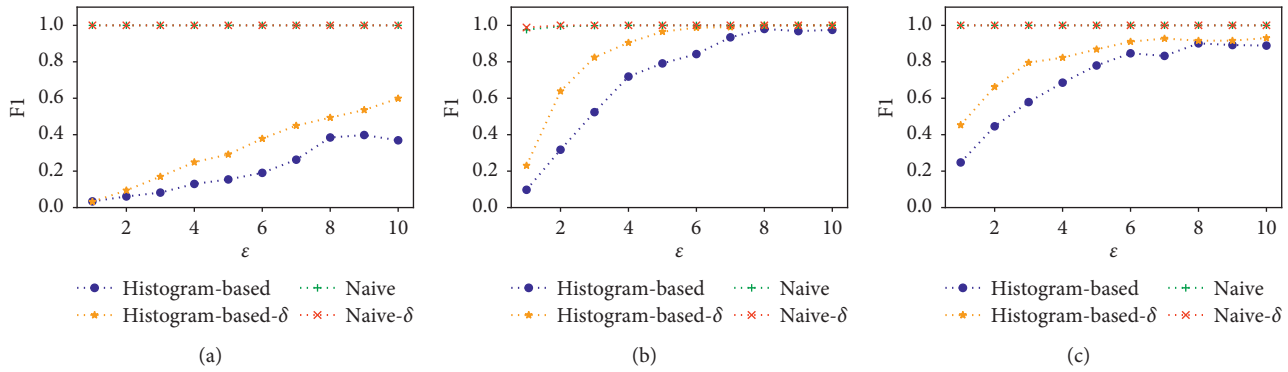

Figure 9: F_1 result for different ε in all 3 monitored LANs.

ARP packets, their absence does not affect privacy or utility to our approaches.

9.2. DP Mechanisms. In this work, we focus on releasing ARP-degrees in differentially-private manners. Publishing degrees has sensitivity of 1 (removing a user's ARP request alters the total ARP-degrees by 1), which is small compared to the number of ARP requests sent by all users. Thus we choose the noise perturbation methods, namely the Laplace and the Gaussian mechanism, to privatize the ARP-degrees. Another well-known differential privacy mechanism is the randomized response, whose standard deviation is $O(\sqrt{N}/\varepsilon)$ [33], which is worse than the standard deviation of the Laplace and Gaussian mechanism, which is $O(1/\varepsilon)$. There are also differential privacy mechanisms based on data synthesis [34]. However, as anomaly detection algorithms look for "spiking" behaviors at a particular time interval, these data synthetic approaches, which try to replicate the distribution of the data as a whole, will not be able to retain the spikes as well as the perturbation mechanisms.

9.3. Time Interval. In our evaluation, we consider the time interval for ARP-data collection to be in a unit of a week. Albeit a bit long, this design choice is necessary as it allows us to incorporate all data (which spans for 30 weeks) into our analysis with higher utility rate and without losing too much privacy budget.

To illustrate this point, we conduct a new experiment on the JPN network where we aggregate and process ARP data on a shorter period, i.e., every day instead of every week. Compared to the original experiment, we have observed a drastic decrease in the utility rate for all our approaches. As an example, for the naïve approach with $\varepsilon = 4$, the RMSE has increased by a factor of 6 (from 10 to 60), while the TPR and F_1 score have reduced substantially from 1.0 to ≈ 0.6.

9.4. Utility Metrics. We evaluate our approaches using two utility metrics: RMSE and Anomaly Detection Accuracy. We select the former because it is one of the most common metrics for measuring utility from a DP mechanism [35]. Intuitively, it tells us "how far apart the privatized data is from the original data." Since an anomalous activity appears as an unusual value in the data, a privacy-preserving mechanism with small RMSE would not perturb *that value* by much, allowing such activity to be detected from the privatized data. Besides RMSE, there are other similar metrics with the same purpose, e.g., Mean Absolute Error. Even though we do not include them in this work, we expect the results from such metrics to be in line with our current results.

Nonetheless, the RMSE does not directly indicate the "true" utility in this work since our end goal is to detect LAN anomalies, not minimize error rates. To this end, we choose to include Anomaly Detection Accuracy as our second metric. This metric realistically gives us an idea of how effective our approaches are when performing on a real-world LAN anomaly detector [7].

Finally, we do not consider other utility metrics that target different types of data publication. For example, L_p-Error [36] and Hausdorff Distance [37] are geared towards measuring utility in location privacy protection. Also, information-theoretic metrics [38] require the input to be generated from a probability distribution, which is not the case in this work.

10. Conclusion

This paper presents four 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 can provide node-differential privacy, thus leaking no information about a presence of each individual user. We also propose two alternatives, named naïve-δ and histogram-based-δ, which require even smaller additive noises than their original counterparts in exchange for a small probability that the privacy guarantee will not hold. Feasibility of our approaches is demonstrated via real-world experiments in which we show that, with a reasonable privacy budget value, our approaches yield low errors (< 10 in RMSE) and also preserve more than 75% utility of detecting LAN anomalies.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

The preliminary (and much shorter) version of this manuscript was published in IEEE International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) 2021 [1].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The ASEAN IVO (https://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 (https://www.nict.go.jp/en/index.html). This work was also financially supported by Chiang Mai University, Thailand.

References

[1] N. Rattanavipanon, D. Ponnoprat, H. Ochiai, K. Tantayakul, T. Angchuan, and S. Kamolphiwong, "Releasing ARP data with differential privacy guarantees for LAN anomaly detection," in *Proceedings of the 2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, IEEE, Chiang Mai, Thailand, May 2021.

[2] IFP, "6 Hacks Sure to Defeat Your Firewall (And How to Prevent Them)," 2018, https://www.insightsforprofessionals.com/it/security/hacks-sure-to-defeat-your-firewall.

[3] W. Yan, Z. Zhang, and N. Ansari, "Revealing packed malware," *IEEE Security and Privacy Magazine*, vol. 6, no. 5, pp. 65–69, 2008.

[4] M. Chapple, "The Threat of Ransomware Still Looms Large over Healthcare," 2021, https://healthtechmagazine.net/article/2021/06/threat-ransomware-still-looms-large-over-healthcare.

[5] C. Kern, "95filtering," 2016, https://www.varinsights.com/doc/of-ransomware-bypass-firewalls-email-filtering-0001.

[6] V. Networks, "How is the internet of things (IoT) being impacted by malware?," 2021, https://www.valeonetworks.com/how-is-the-internet-of-things-iot-being-impacted-by-malware/.

[7] K. Matsufuji, S. Kobayashi, H. Esaki, and H. Ochiai, "Arp request trend fitting for detecting malicious activity in lan," in *Proceedings of the International Conference on Ubiquitous Information Management and Communication (ICUIMC)*, Phuket, Thailand, January 2019.

[8] Y. Yasami, M. Farahmand, and V. Zargari, "An arp-based anomaly detection algorithm using hidden Markov model in enterprise networks," in *Proceedings of the International Conference on Systems and Networks Communications (ICSNC)*, IEEE, Cap Eterel, France, August 2007.

[9] H. Ren, B. Xu, Y. Wang et al., "Time-series anomaly detection service at microsoft," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Alaska, USA, August 2019.

[10] M. Mobilio, M. Orrù, O. Riganelli, A. Tundo, and L. Mariani, "Anomaly detection as-a-service," in *Proceedings of the IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, Berlin, Germany, October 2019.

[11] D. Yao, X. Shu, L. Cheng, and S. J. Stolfo, "Anomaly detection as a service: challenges, advances, and opportunities," *Synthesis Lectures on Information Security, Privacy, and Trust*, vol. 9, no. 3, pp. 1–173, 2017.

[12] M. Azure, "Anomaly Detector," 2020, https://azure.microsoft.com/en-us/services/cognitive-services/anomaly-detector/.

[13] Tibco Software, "Anomaly Detection—Tibco Software," 2021, https://www.tibco.com/solutions/anomaly-detection.

[14] J. Hu, C. Lin, and X. Li, "Relationship privacy leakage in network traffics," in *Proceedings of the 2016 25th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–9, IEEE, Waikoloa, HI, USA, August 2016.

[15] M. Srivatsa and M. Hicks, "Deanonymizing mobility traces: using social network as a side-channel," in *Proceedings of the ACM Conference on Computer and Communications Security*, Republic of Korea, November 2012.

[16] N. Takbiri, A. Houmansadr, D. L. Goeckel, and H. Pishro-Nik, "Privacy against statistical matching: inter-user correlation," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Vail, CO, USA, June 2018.

[17] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the Theory of Cryptography Conference (TCC)*, New York, USA, March 2006.

[18] F. McSherry and R. Mahajan, "Differentially-private network trace analysis," *ACM SIGCOMM-Computer Communication Review*, vol. 40, no. 4, pp. 123–134, 2010.

[19] F. D. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *Proceedings of the International Conference on Management of Data*, Rhode Island, USA, July 2009.

[20] L. Fan, L. Bonomi, L. Xiong, and V. Sunderam, "Monitoring web browsing behavior with differential privacy," in *Proceedings of the International Conference on World Wide Web (WWW)*, Seoul, Korea, April 2014.

[21] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren, "Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 591–606, 2016.

[22] F. K. Dankar and K. El Emam, "Practicing differential privacy in health care: a review," *Transactions on Data Privacy*, vol. 6, no. 1, pp. 35–67, 2013.

[23] L. Fan and L. Xiong, "Differentially private anomaly detection with a case study on epidemic outbreak detection," in *Proceedings of the International Conference on Data Mining Workshops (ICDMW)*, Dallas, TX, USA, December 2013.

[24] Z. Zhang, H. Esaki, and H. Ochiai, "Unveiling malicious activities in lan with honeypot," in *Proceedings of the International Conference on Information Technology (InCIT)*, Bangkok, Thailand, October 2019.

[25] J. Yeo, M. Youssef, and A. Agrawala, "A framework for wireless lan monitoring and its applications," in *Proceedings of the 3rd ACM Workshop on Wireless Security*, New York, USA, October 2004.

[26] D. Whyte, P. van Oorschot, and E. Kranakis, "Arp-based detection of scanning worms within an enterprise network," in *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, Tucson, AZ, USA, December 2005.

[27] M. Farahmand, A. Azarfar, A. Jafari, and V. Zargari, "A multivariate adaptive method for detecting arp anomaly in local area networks," in *Proceedings of the International Conference on Systems and Networks Communications (ICSNC)*, Tahiti, French Polynesia, October 2006.

[28] M. Bun and T. Steinke, "Concentrated differential privacy: simplifications, extensions, and lower bounds," in *Proceedings of the Theory of Cryptography Conference (TCC)*, Beijing, China, October 2016.

[29] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[30] M. Hay, C. Li, G. Miklau, and D. D. Jensen, "Accurate estimation of the degree distribution of private networks," in *Proceedings of the ICDM 2009, the Ninth IEEE International Conference on Data Mining*, Miami, FL, USA, December 2009.

[31] I. Mironov, "On significance of the least significant bits for differential privacy," in *Proceedings of the ACM Conference on Computer and Communications Security*, North Carolina, USA, October 2012.

[32] D. C. Montgomery and C. M. Mastrangelo, "Some statistical process control methods for autocorrelated data," *Journal of Quality Technology*, vol. 23, no. 3, pp. 179–193, 1991.

[33] A. Beimel, K. Nissim, and E. Omri, "Distributed private data analysis: simultaneously solving how and what," in *Proceedings of the Advances in Cryptology-CRYPTO 2008, 28th Annual International Cryptology Conference*, Santa Barbara, CA, USA, August 2008.

[34] Y. Tao, R. McKenna, M. Hay, A. Machanavajjhala, and G. Miklau, "Benchmarking differentially private synthetic data generation algorithms," 2021, https://arxiv.org/pdf/2112.09238.

[35] X. Yang, T. Wang, X. Ren, and W. Yu, "Survey on improving data utility in differentially private sequential data publishing," *IEEE Transactions on Big Data*, vol. 7, 2017.

[36] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1298–1309, New York, USA, October 2015.

[37] J. Hua, Y. Gao, and S. Zhong, "Differentially private publication of general time-serial trajectory data," in *Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM)*, IEEE, Hong Kong, China, April 2015.

[38] M. Lopuhaä-Zwakenberg, B. Škorić, and N. Li, "Information-theoretic Metrics for Local Differential Privacy Protocols," 2019, https://arxiv.org/pdf/1910.07826.