electronics

MDPI

Article

PD-PAn: Prefix- and Distribution-Preserving Internet of Things Traffic Anonymization

Xiaodan Gu * and Kai Dong

School of Computer Science and Engineering, Southeast University, Nanjing 211189, China
* Correspondence: xdgu@seu.edu.cn

Abstract: One of the features of network traffic in Internet of Things (IoT) environments is that various IoT devices periodically communicate with their vendor services by sending and receiving packets with unique characteristics through private protocols. This paper investigates semantic attacks in IoT environments. An IoT semantic attack is active, covert, and more dangerous in comparison with traditional semantic attacks. A compromised IoT server actively establishes and maintains a communication channel with its device, and covertly injects fingerprints into the communicated packets. Most importantly, this server not only de-anonymizes other IPs, but also infers the machine states of other devices (IPs). Traditional traffic anonymization techniques, e.g., *Crypto-PAn* and *Multi-View*, either cannot ensure data utility or is vulnerable to semantic attacks. To address this problem, this paper proposes a prefix- and distribution-preserving traffic anonymization method named *PD-PAn*, which generates multiple anonymized views of the original traffic log to defend against semantic attacks. The prefix relationship is preserved in the real view to ensure data utility, while the IP distribution characteristic is preserved in all the views to ensure privacy. Intensive experiments verify the vulnerability of the state-of-the-art techniques and effectiveness of *PD-PAn*.

Keywords: Internet of Things; traffic anonymization; semantic attack; *Crypto-PAn*

check for updates

Citation: Gu, X.; Dong, K. *PD-PAn*: Prefix- and Distribution-Preserving Internet of Things Traffic Anonymization. *Electronics* **2023**, *12*, 4369. https://doi.org/10.3390/electronics12204369

Academic Editors: Rameez Asif, Yue Zhang, Yinghui Zhang, Gang Han and Ming Li

Received: 10 September 2023
Revised: 13 October 2023
Accepted: 17 October 2023
Published: 210 October 2023

1. Introduction

As network traffic analysis (NTA) grows more sophisticated, there is an increasing need for organizations to outsource NTA tasks to third-party analysts. NTA uses network traffic data to detect and investigate security threats and anomalous or malicious behaviors within that network [1]. According to research from ESG [2], 87% of organizations use NTA tools for threat detection and response today. Recent NTA services (e.g., Auvik [3], MixMode [4]) make use of deep learning techniques to provide comprehensive security monitoring and analysis functionalities based on the cloud. In contrast with the convenience and effectiveness achieved by using cloud-based NTA services, organizations have to upload their network traffic data, thus raising significant privacy concerns.

IP anonymization techniques are typically used to preserve privacy in network traffic logs. In NTA scenarios, the prefix relationship has to be preserved to ensure data utility. *Crypto-PAn* [5] is the most widely used prefix-preserving IP anonymization technique. However, it is vulnerable to semantic attacks [6,7]. A semantic attack is a combination of an injection attack and a fingerprinting attack. The adversary passively waits for a target user to initiate a communication, injects fingerprints into the traffic, identifies the traffic in the anonymized log, and de-anonymizes all IP addresses with the same prefix as an IP in the identified traffic. *Multi-View* [8] is a defense against semantic attacks. It generates multiple fake traffic logs to conceal the real one. Due to this reason, even if the adversary identifies the injected traffic and de-anonymizes the IPs within the traffic, they cannot de-anonymize the other IPs. Thus, privacy is preserved.

However, Internet of Things (IoT) network traffic has its unique feature, which raises new challenges in traffic anonymization. In this paper, we propose an IoT semantic attack that is active, covert, and more dangerous in comparison with a traditional semantic attack. In an IoT environment, various IoT devices periodically communicate with their vendor services. Thus, an adversarial or compromised server can perform injection and fingerprinting attacks more covertly. Moreover, since the traffic generated by IoT devices can be used to detect anomalies [9–11] and infer changes of the machine states [12–14], the adversary who performs an IoT semantic attack not only de-anonymizes IPs of IoT devices, but also infers private IoT events.

In contrast with the active, covert, and more dangerous IoT semantic attack, the state-of-the-art defense (i.e., *Multi-View*) against semantic attacks suffers from significant drawbacks. First, it suffers from poor data utility since the anonymization applied in *Multi-View* causes a collapsed prefix domain and a destroyed prefix relationship. Second, *Multi-View* is vulnerable to de-anonymization attacks since the unique distribution characteristic within the real traffic log makes it distinguishable from other fake traffic logs.

In this paper, we propose *PD-PAn*, a traffic anonymization method that preserves prefix relationships to ensure data utility while preserving IP distribution characteristics to defend against semantic attacks. At the idea level, an IP anonymization operation divides the IP prefix domain into several disjoint prefix sets. Each set can be abstracted as a prefix ring, where each node on the ring is a prefix, and the next node can be obtained by calculating the anonymized form of this prefix. To resist collapse of the prefix domain, *PD-PAn* specifies parameters for the anonymization operation. Based on the prefix ring, *PD-PAn* is able to generate a real view of the traffic log, which is prefix-preserving, by simply using the prefix-preserving IP anonymization operation proposed in *Cypto-PAn*. After that, *PD-PAn* has to construct multiple fake views to conceal the real view.

It is non-trivial to construct fake views that are indistinguishable from real views. Fake views should neither be too similar nor too different from real views. On the one hand, one can generate some views that are similar to the real view by simply using a prefix-preserving IP anonymization operation again on the real view. However, views generated in this way are, in essence, real views. Although they have different IPs in the traffic, the prefix relationship among IPs is totally the same between the two views. Due to this reason, analysts will generate the same report on different views. Moreover, if an adversary performs a semantic attack, the de-anonymization results in different views will be the same. Thus, all these views are still vulnerable to such attacks. On the other hand, one can randomly construct some views so that these views are different from real views (as in *Multi-View*). However, views constructed in this way can be easily identified based on their IP distribution characteristics. *PD-PAn* first uses a sampling algorithm without replacement to ensure that different IP prefixes can be anonymized to the same result in a fake view. It then uses the *Fisher–Yates shuffle* algorithm [15] to ensure the IP distribution in the real view and that in any fake view are indistinguishable.

We conduct experiments on both a public dataset and a real-world private dataset to evaluate privacy achieved by *PD-PAn* in terms of trace anonymity, and evaluate secrecy in terms of the averaged number of leaked bits in IP addresses. Under the premise of prefix-preserving, *PD-PAn* achieves an adequate level of privacy and secrecy. Even compared with state-of-the-art defense against semantic attacks that are not prefix-preserving (i.e., *Multi-View*), *PD-PAn* still wins in both privacy and secrecy.

To summarize, this paper makes the following contributions:

- We propose *PD-PAn*, which is the first traffic anonymization method that is both prefix-preserving and resistant to semantic attacks.
- We propose and investigate an IoT semantic attack that is active, covert, and more dangerous than traditional semantic attacks.
- We conduct intensive experiments to validate the advancements of *PD-PAn*.

The rest of this paper is organized as follows. Section 2 introduces the background related to this paper. In Section 3, we describe the adversary model, propose an IoT semantic attack, and provide some insights into this attack. We then propose a novel method *PD-PAn* in Section 4. We perform a security analysis in Section 5. Section 6 evaluates *PD-PAn*, and Section 7 gives a brief survey on related techniques. Section 8 concludes this paper.

2. Background

We consider scenarios where network traffic logs in an IoT environment need to be transferred by the data owner, e.g., for security analysis outsourcing [8]. IP anonymization techniques are always used to preserve privacy; however, they are vulnerable to semantic attacks. In this section, the background of related techniques is introduced, and the notations used throughout this paper are summarized in Table 1.

Table 1. Summary of Notations.

Symbol	Meaning
$PP(), RPP()$	Prefix-preserving operation of *Crypto-PAn*, reverse operation of PP
L_o, L, L_R, L_F, L_S	Original traffic log, encrypted log, real view, fake view, seed view
$\mathcal{K}_1, \mathcal{K}_2$	Key for generating encrypted log L, key for generating real view and fake views
d_I, d_P, b_P	Number of distinct IP addresses, number of distinct IP prefixes, number of bits in an IP prefix
$m, \mathcal{G}, c$	Prefix mask, prefix ring, group size used for prefix grouping
α, β	Proportion of injected IP prefixes, number of injected IP addresses for each injected prefix

2.1. Crypto-PAn

Crypto-PAn [5] is a cryptography-based, prefix-preserving IP anonymization technique. For a given n-bit IP address $x = x_1x_2\cdots x_n$, let $\mathcal{K}$ be a private key specified by the user. The anonymizing function $PP(\cdot,\cdot)$ is defined as

$$PP(x, \mathcal{K}) = x'_1x'_2\ldots x'_n.$$

where $x'_i = x_i \oplus f_{i-1}(x_1, x_2, \ldots, x_{i-1})$, f_0 is a constant, f_i is a function from $\{0,1\}^i$ to $\{0,1\}$ $(i = 1, 2, \cdots, n-1)$, and $\oplus$ represents the XOR boolean operator,

$$f_i = \mathcal{L}(\mathcal{R}(\mathcal{P}(x_1x_2\ldots x_i)), \mathcal{K}), (i = 1, \ldots, n-1). \quad (1)$$

where $\mathcal{L}(\cdot)$ returns the least significant bit, $\mathcal{R}(\cdot)$ is a block cipher, and $\mathcal{K}$ is the key used in $\mathcal{R}$, $\mathcal{P}(\cdot)$ is a padding function that expands $x_1x_2\cdots x_i$ into a longer string that matches the block size of $\mathcal{R}$. The anonymizing function PP can be used in an iterative fashion. For the n-bit IP address x and the key $\mathcal{K}$, $PP^j(x, \mathcal{K})$ implies that x is anonymized j times by using key $\mathcal{K}$.

One important feature of *Crypto-PAn* is prefix-preserving.

Definition 1 (Prefix-Preserving)**.** *For any given two IP addresses x and y, and any given key $\mathcal{K}$, the anonymized IP addresses $PP^j(x, \mathcal{K})$ and $PP^j(y, \mathcal{K})$ satisfy: if and only if x and y share a k-bit prefix, $PP^j(x, \mathcal{K})$ and $PP^j(y, \mathcal{K})$ share a k-bit prefix, for any $j \in \mathcal{N}$ and any $k \in [0, n]$.*

An original IP address can be recovered from its anonymized form. This process is known as reverse prefix-preserving (RPP), which can be formalized as a function RPP:

$$RPP(PP(x, \mathcal{K}), \mathcal{K}) = x.$$

where x is an arbitrary IP address, and $\mathcal{K}$ is an arbitrary key. RPP is also prefix-preserving. When extending to the case of anonymization multiple times (i.e., PP^j), the original IP address can be recovered by performing RPP the same number of times (i.e., $RPP^j = PP^{-j}$).

2.2. Traditional Semantic Attack

A traditional semantic attack can be performed by an adversarial server to compute true IP addresses from anonymized traces. It is composed of three phases.

Phase 1: Passive Communication. In the first phase, the adversary waits for a target user to initiate a communication request. Then, a communication channel is established.

Phase 2: Injection and fingerprinting. In the second phase, the adversary sends packets to the target in some unusual pattern to inject so-called quasi-identifiers (i.e., fingerprints, including specific time, port, protocol, or other attributes) into the traffic [6,16,17].

Phase 3: De-Anonymizing. In the third phase, the adversary tries to acquire as many real IPs as possible. They first obtain an anonymized traffic log. Within this log, they identify the injected traffic and obtain a mapping from the prefix of an anonymized IP to that of the injected IP. Then, by using this mapping, they de-anonymize all IP addresses that share the same prefix with the injected one.

2.3. Multi-View as a Defense

Multi-View [8] is a defense against traditional semantic attacks. Its core idea is to construct and make use of multiple indistinguishable views of traffic logs, including a real view and multiple fake views. In different views, the anonymized forms of the same IP address are different; moreover, the same anonymized IP address can also be encrypted from different original IPs. Due to this reason, the prefix relationship is totally destroyed in both the real view and the fake view. Even if the adversary performs a semantic attack, they can only de-anonymize the injected IP but not the other IPs.

Multi-View is mainly composed of three steps:

Step 1: Initialization. The data owner specifies two keys, denoted as $\mathcal{K}_1$ and $\mathcal{K}_2$. $\mathcal{K}_1$ is used to encrypt the original traffic log denoted as L_o. We use L to denote the encrypted log, which is generated by using the PP operation:

$$L = PP(L_o, \mathcal{K}_1). \tag{2}$$

L_o and $\mathcal{K}_1$ are no longer used to ensure their secrecy.

In subsequent anonymizing operations, only L and $\mathcal{K}_2$ are used.

Step 2: Real View Generation. The data owner specifies a k-bit prefix mask m (and the inverse mask of m is $2^n - m - 1$ in binary), sets up a group for each k-bit prefix, and moves all IP addresses with the same prefix in this group. After that, she randomly specifies an order number for each group. Then, for each IP address, the data owner lines up the IP and the inverse mask of m together to obtain a transformed IP (the prefix of which is always 0), and anonymizes the transformed IP by performing the PP operation j times where j is the order number of the group in which the IP belongs. After all IP addresses are anonymized, the real view is generated.

Step 3: Fake View Construction. The process of constructing a fake view is similar to that of generating a real view, except that all IP addresses are grouped randomly. Multiple fake views are generated so as to conceal the real view.

2.4. Event Inference Attack in IoT

An IoT event inference attack can be performed to infer sensitive information about the target IoT device by statistically analyzing its non-sensitive information. For example, to perform an IoT event inference attack, an adversary first eavesdrops on the encrypted traffic sent to/from an IoT device, then analyzes external attributes, e.g., packet length [12,18] of the encrypted traffic, and determines the traffic fingerprints generated with certain IoT events. It has already been verified by existing research that the traffic generated by an IoT device can be used to infer an IoT event (e.g., change of the machine state) [12–14,18–22]. Another kind of IoT event inference attack classifies IoT traffic [23,24] based on machine learning techniques.

3. Semantic Attack on Anonymized IoT Traces

In this section, we introduce the IoT semantic attack, highlight the difference between this attack and the traditional semantic attacks, and provide some insights into this attack.

3.1. Attack Model

The original traffic log L_o is a collection of network traces, and each trace contains multiple fields, including IP addresses, port numbers, timestamps, etc. IPs are identifiers, and other fields are quasi-identifiers, i.e., the fingerprints. L_o should be anonymized/encrypted before being transferred to a third-party analyst to protect privacy. However, an adversarial or compromised analyst is still able to de-anonymize the IPs by making use of the fingerprints. The adversary model considered in this paper is in commitment with previous research [5,6,16,17]. The assumptions are as follows:

1. The analyst is honest but curious. After they obtain the anonymized IoT traces transferred from the data owner, they will honestly perform security analysis and return the analysis report, but also will try to de-anonymize as many IPs as possible in the anonymized traces.
2. The prefix-preserving function $PP()$ is public, but all the keys used in this function can be kept as secret.
3. The communication channel between the data owner and the analyst is secure, that is, issues related to integrity and availability are not taken into consideration.

3.2. IoT Semantic Attack

In an IoT environment, various IoT devices periodically communicate with their vendor services to report an event, inform the status, receive a command, upload collected data, or send/receive an ACK, etc. Such communication is always based on a private protocol specified by the vendor service. Due to this reason, a semantic attack on IoT network traces is active, covert, and more dangerous in comparison with a traditional semantic attack. An IoT semantic attack is composed of three phases.

Phase 1: Active Communication. In the IoT scenario, an adversarial vendor service no longer needs to passively wait for the establishment of a communication channel. Instead, it simply uses the communication channel maintained by the IoT device and itself. Such channels are essential for IoT functionalities like remote controlling, and are established and maintained based on a private protocol, which is specified by the service itself.

Phase 2: Injection and Fingerprinting. The IoT devices communicate with the server continuously and frequently, sending/receiving messages with distinct semantic characteristics. An adversary can therefore perform an injection and fingerprinting attack more conveniently and more covertly by using its private protocol.

Phase 3: De-Anonymizing and Inference. A traditional semantic attack can only de-anonymize the IP address of an entity in a network that shares the same prefix with the injected IP, thereby revealing the original network traces containing that IP, but still cannot obtain the content of the traffic. However, for the IoT traffic logs, an adversary is able to distinguish traffic features of different IoT events and perform an event inference attack to infer the content of the traffic (e.g., an IoT event) so as to obtain the private usage pattern of all IoT devices with the same IP prefix.

3.3. Limitations of Existing Defense

Drawback 1: Poor Data Utility.

Multi-View suffers from poor data utility (even in the real view) due to the following two reasons.

Problem 1: Collapsed Prefix Domain.

In *Multi-View*, the number of distinct prefixes in all the views is always smaller than that in the original traffic log. This is because different prefixes are anonymized by performing the PP operation at different times on the same transformed IP prefix (which is set to be 0). If the results generated are the same, then the anonymized prefixes of different original prefixes are also the same. In this case, the prefix domain is collapsed. Consider the size of a prefix domain is 2^k. There is no guarantee that there are also 2^k different results of $PP^j(0, \mathcal{K}_2), \forall j \in \mathcal{N}$. Further consider a key $\mathcal{K}'$ which satisfies $PP(0B0000, \mathcal{K}_2) = 0B0000$, the size of the prefix domain is reduced from 2^4 to 1. Then, there is only 1 prefix in all the views; however, there can be up to 16 prefixes in the original traffic log. Due to this reason, the analyst cannot distinguish all IP prefixes when performing security analysis.

Problem 2: Destroyed Prefix Relationship. The hierarchical (prefix) relationship [25] plays an important role in NTA. However, *Multi-View* fails to preserve this relationship even in the real view. Any IP address is first lined up with the inverse of the prefix mask m to transform to an IP with prefix 0, and is then anonymized by using a PP operation several times. Consider two IPs that have the same top-k bits but their prefixes are different, so they are in different groups. Further, consider the order numbers of their groups are j_1 and j_2, respectively. Then, the prefixes of the anonymized IPs are $PP^{j_1}(0, \mathcal{K}_2)$ and $PP^{j_2}(0, \mathcal{K}_2)$. Since $j_1 \neq j_2$, there is no guarantee that the two anonymized IPs still have top-k' bits in common. This means that the prefix relationship among the IP addresses is totally destroyed. Therefore, the data utility of the generated views (including the real view) is very poor.

Drawback 2: Vulnerable Anonymization. *Multi-View* is vulnerable when applied to real-world network traffic since the generated real views and fake views can be distinguished due to the following reason.

Problem 3: Unique Distribution Characteristic.

In *Multi-View*, the real view can be identified by simply calculating a cumulative density function (CDF) over top-d prefixes in population on different d. The CDF in the real view is the same as that in the original traffic log (if the prefix domain is not collapsed), but is very different from that in fake views as illustrated in Figure 1. The difference between the CDFs in two fake views is almost negligible, while that between a real view and a fake view is significant. This is because a real-world traffic log has its unique distribution of IPs while the anonymized IPs in fake views are randomly generated and evenly distributed. Even if one improves *Multi-View* by using different distributions (e.g., normal distributions) to construct fake views, an adversary can still identify a real view due to its unique distribution characteristic.

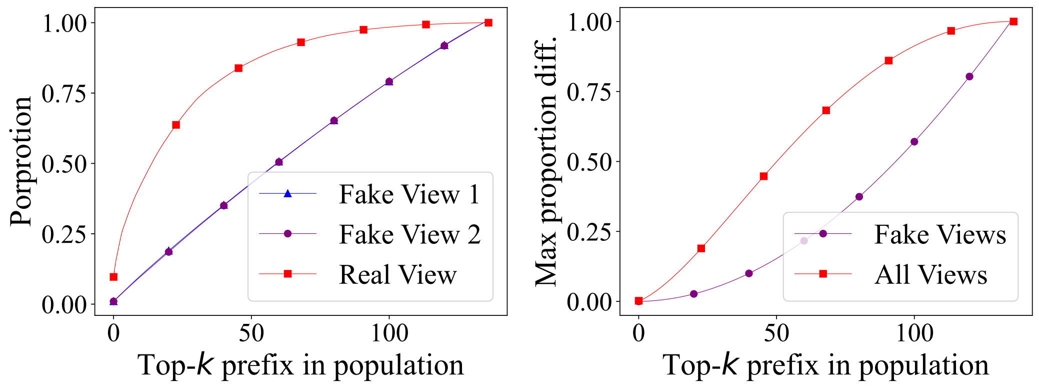

Figure 1. In *Multi-View*, the real view is identifiable based on its unique distribution of IPs on top rank prefixes in population.

3.4. Goal

The main goal of this paper is to provide a defense against semantic attacks while ensuring data utility (i.e., prefix-preserving) in the traffic logs transferred to the analysts. It has already been verified that any prefix-preserving IP anonymization algorithm is vulnerable to semantic attacks [6,7]. Due to this reason, the defense should also generate multiple views. On the one hand, the prefix relationship is preserved in the real view to ensure data utility; on the other hand, the fake views are indistinguishable from the real view to ensure privacy against semantic attacks.

4. Methodology

In this section, we propose a prefix- and distribution-preserving IP anonymization method named *PD-PAn*.

4.1. Overview

PD-PAn addresses the three problems discussed in Section 3.3. The basic idea to address each problem is introduced as follows.

4.1.1. Idea 1: Collapse Resistant Key Selection

To ensure the data utility in the real view, the size of the prefix domain needs to be maintained. The cryptography used in the PP operation in *Multi-View* is the key reason for the prefix domain collapsing. We find that the key $\mathcal{K}_2$ used to generate multiple views determines whether a prefix domain will collapse. For any given key and an IP prefix, we can construct a prefix ring, where each node on the ring is a prefix, and the next node can be obtained by calculating the anonymized prefix by using PP once to that node. Due to this reason, our basic idea of *PD-PAn* to resist collapse in the prefix domain is to select an appropriate key without reducing security to construct a prefix ring that holds enough space for every IP prefix in the original log.

4.1.2. Idea 2: Prefix-Preserving Real View Generation

The real view should preserve the prefix relationship. In *Multi-View*, the prefix relationship is destroyed in the step of transforming IP addresses by lining up each IP with the inverse of a prefix mask m. Due to this reason, our basic idea of *PD-PAn* to preserve prefix relationships is straightforward, simply using prefix-preserving techniques, e.g., a PP operation, in a reasonable way.

4.1.3. Idea 3: Distribution-Preserving Fake View Construction

To construct fake views that are indistinguishable from the real view, we have to ensure that the IP distribution in all the views is unidentifiable. We consider two simple functions that preserve the distribution of variables: the sampling function and the shuffling function. The basic idea of *PD-PAn* to preserve the IP distribution is to combine the PP operation and the two functions. We generate a number for each IP to determine how many times the PP operation is performed by using the sampling function. Then, we use the shuffling function to generate multiple orders, each of which represents a fake view. This method ensures that the IP distribution characteristics in the generated logs are the same as those in the original logs, thus avoiding the security risks caused by the IP distribution characteristics.

4.1.4. High-Level Strategy

The high-level strategy of *PD-PAn* is to generate a real view and multiple fake views of a traffic log. The goal of *PD-PAn* is two-fold. On the one hand, data utility, including prefix relationships in the real view, has to be ensured. On the other hand, the indistinguishability of the real view from the fake views should also be guaranteed. *PD-PAn* uses a three-step procedure as illustrated in Figure 2 to achieve this goal. In the first step, two keys are generated, where one of them is carefully selected so that *PD-PAn* is able to construct a prefix ring that holds enough space for each distinct IP prefix. In the second step, the real view is generated by simply using a PP operation on the encrypted log so that the prefix relationship is preserved. In the third step, fake views are generated by jointly utilizing a sampling function, a shuffling function, and the PP operation, to ensure the indistinguishability of IP distributions in these views. By following an improved protocol, multiple views can be generated at the side of the analyst to reduce transmission overhead. This protocol is detailed in Section 4.3.

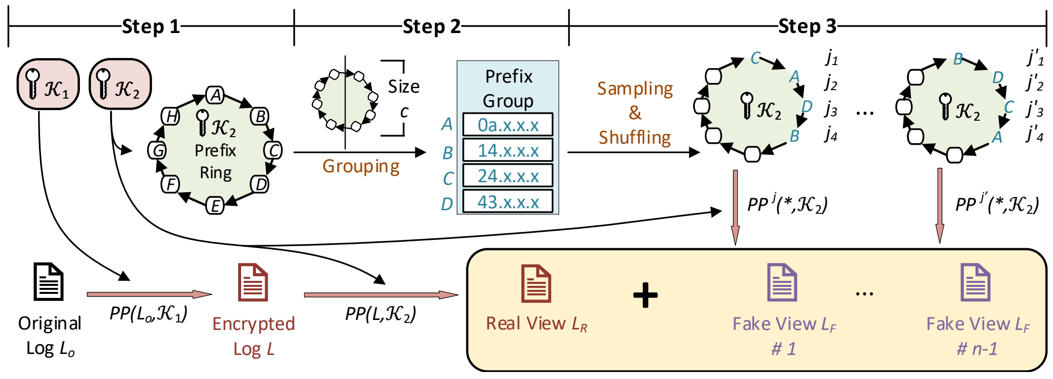

Figure 2. Overview of *PD-PAn*.

4.2. Details

4.2.1. Step 1: Collapse Resistant Initialization

In this step, *PD-PAn* generates two keys: $\mathcal{K}_1$ and $\mathcal{K}_2$. $\mathcal{K}_1$ is randomly generated. With $\mathcal{K}_1$, *PD-PAn* encrypts the original log L_o to generate an encrypted log L by using the *Crypto-PAn* algorithm (i.e., the *PP* operation).

$\mathcal{K}_2$ is randomly picked by *PD-PAn* from a key pool that is constructed for each predefined prefix mask m. We would like to note that $\mathcal{K}_2$ is not completely random generated since it should be carefully selected or examined to meet the requirement that the number of distinct prefixes in an anonymized view equals that of prefixes in the encrypted log.

For an IP address x and a prefix mask m, *PD-PAn* can construct for any given key $\mathcal{K}$ a prefix ring denoted as $G_\mathcal{K}$ to verify whether $\mathcal{K}$ can be selected. $G_\mathcal{K}$ is constructed as follows. Let j_{min} be the minimum positive integer satisfying that the prefix of IP x equals the prefix of the IP generated by anonymizing x for j_{min} times, given by:

$$(PP^{j_{min}}(x,\mathcal{K})\&m) = (x\&m). \tag{3}$$

where $(\cdot\&m)$ calculates the prefix of an IP by lining up this IP and m. *PD-PAn* obtains j_{min} anonymized prefixes by calculating:

$$(PP^{j}(x,\mathcal{K})\&m), \tag{4}$$

for all $j \in \mathcal{N}, 0 \le j < j_{min}$. Let each anonymized prefix be a node, the anonymization relation between two nodes as edges, and the prefix ring $G_\mathcal{K}$ of size j_{min} is constructed. An example prefix ring of size $j_{min} = 8$ is illustrated in Figure 3.

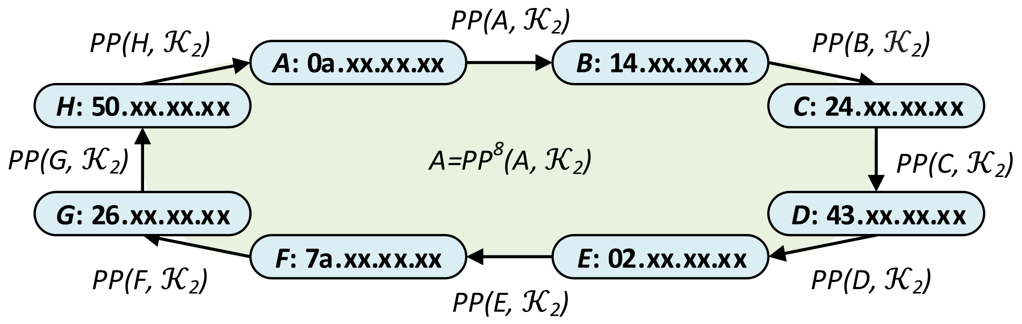

Figure 3. An example IP prefix ring composed of 8 prefixes.

Let d_L be the number of prefixes in the encrypted traffic log L. *PD-PAn* has to ensure d_L is no greater than the size of the prefix ring j_{min} to resist collapse in the prefix domain (we will later relax this constraint, detailed in Section 5.2), given by:

$$d_L \le j_{min}. \tag{5}$$

PD-PAn generates in advance many random keys to compose a key pool, and computes and records for each key the size of its prefix ring. We would like to note that for a given key, there can be several prefix rings with different sizes. *PD-PAn* only records the size of the ring containing the prefix of IP 0.0.0.0, and uses this ring as the default ring. In generating $\mathcal{K}_2$, *PD-PAn* only needs to select a key from the key pool, and guarantees that the size of the default ring constructed with this key is large enough.

PD-PAn is always able to find a key that meets the requirement in (5). Let b_P denote the number of bits in an IP prefix (which equals the number of 1 s in m), we have that the maximum value of j_{min} should be 2^{b_P}. When $j_{min} = 2^{b_P}$, the prefix ring $G_{\mathcal{K}}$ is a full ring. In this case, the inequality in (5) always holds. After this step, $\mathcal{K}_1$ and L_o are never used in *PD-PAn* and kept secret. $\mathcal{K}_2$ and L are used as the parameters in subsequent steps. To facilitate the understanding of *PD-PAn*, now, we consider $\mathcal{K}_2$ and L are also kept secret. Later in Section 5, we will explain why they can be set to be public in our improved protocol without reducing security.

4.2.2. Step 2: Prefix-Preserving Real View Generation

In this step, *PD-PAn* generates a real view denoted as L_R within which the prefix relationship among IPs is preserved. *PD-PAn* simply makes use of the PP operation, which itself is a prefix-preserving anonymization function.

There is still a problem dealing with computational overhead. In the subsequent step, *PD-PAn* still needs to construct multiple fake views to conceal the real one. Due to this reason, the PP operation will be performed multiple times to anonymize each IP. To reduce computational overhead, *PD-PAn* specifies a parameter c to denote an upper bound on the number of times that the PP operation is performed on an IP address. *PD-PAn* uses a grouping strategy to put all IP prefixes in different groups by specifying a step number equals c on the prefix ring. Within each group, there are at most c distinct IP prefixes that are close to each other in the prefix ring, like the prefixes A, B, C, D in the figure. This means that an IP prefix can be anonymized to any other prefix within the same group, by performing PP operation (or RPP operation) at most c times.

4.2.3. Step 3: Distribution-Preserving Fake View Construction

In this step, *PD-PAn* constructs n-1 fake views. The IP distribution characteristic of the encrypted log L should be preserved in each fake view, so that all fake views and the real view are indistinguishable with each other. Let d_P denote the number of all IP prefixes, j_{min} denote the size of $G_{\mathcal{K}}$. *PD-PAn* uses the following procedure to construct a fake view:

1. For each prefix (denoted as p) in the encrypted log L, *PD-PAn* counts the number (denoted as d_p) of distinct IPs with this prefix. Then, *PD-PAn* randomly generates an anonymized IP prefix (denoted as p') by using *sampling without replacement* from all the prefixes in the same group, and puts the pair (p', d_p) in an array. Finally, there are d_P pairs in the array.
2. For each anonymized prefix p', *PD-PAn* randomly selects d_p distinct IP addresses in the same group. *PD-PAn* uses the *Fisher–Yates shuffle* algorithm [15] to accomplish the assignments for a whole group all at once. Finally, each distinct IP address in L is assigned an anonymized prefix.
3. For each distinct IP address x, calculate a number j satisfying that the assigned prefix can be obtained by performing the PP operation j times (or by performing the RPP operation $-j$ times if $j < 0$) on x with $\mathcal{K}_2$. After all IPs are anonymized, a fake view is constructed.

4.3. An Improved Protocol

Now we introduce an improved protocol to reduce the transmission overhead required by *PD-PAn*. The basic idea is to rely on the analysts to generate multiple views by themselves. Given any view L', a new view L'' can be generated by applying the PP operation on each IP address several times. Due to this reason, if we record the number of times that PP operation is performed for each IP address in an array, then anyone who has a correct key is able to construct L'' from L' based on this array. Similarly, the real view and all the fake views can also be generated from any given view. *PD-PAn* constructs a seed view denoted as L_S, which is transformed from the encrypted log L by performing the RPP operation on each IP prefix a random number of times (this number is limited to be no greater than c to limit the computational overhead). *PD-PAn* then calculates an array for each view (real view or fake view), satisfying that the view can be constructed from L_S based on this array. After that, *PD-PAn* only needs to transfer the seed view L_S and N arrays to the analysts. The analysts will construct N views including a real view and $N-1$ fake views.

The improved protocol makes the following changes to the three steps in *PD-PAn*.

1. In **Step 1**, *PD-PAn* generates a seed view denoted as L_S. Both $\mathcal{K}_2$ and L_S are considered public.
2. In **Step 2** and **Step 3**, the real view and the fake views are not transferred to the analysts. Instead, *PD-PAn* transfers a key $\mathcal{K}_2$, a seed view L_S and N arrays.

5. Analysis

PD-PAn uses the PP operations several times, but in a different way with *Crypto-PAn* and *Multi-View*. The first difference is in Step 1, where $\mathcal{K}_2$ is not randomly generated but is selected from a key pool. The second difference is in Step 2 and 3, where the PP operations are performed with a public key $\mathcal{K}_2$. We now analyze the security consequence of these two differences, to show that *PD-PAn* does not introduce new vulnerabilities while defending against semantic attacks.

5.1. Analysis on Secrecy of $\mathcal{K}_2$

$\mathcal{K}_2$ is considered public, so the security level is reduced if $PP(\cdot, \mathcal{K}_2)$ is considered as an encryption operation. However, the security level achieved by performing $PP(\cdot, \mathcal{K}_1)$ is not reduced by a public $\mathcal{K}_2$ since $\mathcal{K}_1$ is always kept secret. From this point of view, $PP(\cdot, \mathcal{K}_2)$ only serves as a public transformation operation. Due to this reason, *PD-PAn* is at least as secure as *Crypto-PAn*.

5.2. Analysis on Randomness of $\mathcal{K}_2$

The randomness in generating $\mathcal{K}_2$ is reduced since it is selected from a key pool and it has to satisfy the inequality in (5). We set m = 0xff000000, so there are 256 prefixes. We then randomly generate 1000 keys, so we have 256,000 test cases. We count for each test case the size of the constructed prefix ring. The results are as shown in Table 2. Only 3 keys can be used to construct a full ring. This is the reason the IP domain always collapses by using *Multi-View*. In *PD-PAn*, since the grouping strategy is used, we need to guarantee that the size of the constructed prefix ring on a given IP is no less than the group size c. Otherwise the anonymity level is reduced. If $c = 32$, there is already a chance of 31.75% that the prefix ring, which contains a given prefix, is large enough. Most importantly, although the randomness of $\mathcal{K}_2$ is reduced, this does not influence the security level achieved by *PD-PAn* since $\mathcal{K}_2$ is considered public.

Table 2. Probability of the size of the prefix ring for a random key. The length of a prefix is set to 8 bits.

Size of Prefix Ring	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Probability	0.3%	2.6%	9.73%	19.11%	22.38%	19.25%	12.86%	7.01%	6.77%

6. Evaluation

We conduct experiments on two datasets.

- The `unibs` dataset [26,27]. A public dataset that is collected on campus border routers and is already anonymized by using *Crypto-PAn*. In this dataset, 6483 distinct IP addresses are unevenly distributed in 137 different/8 prefixes.
- The `iot` dataset. A real-world dataset that is collected within a laboratory IoT environment where 27 kinds of home IoT devices from 16 vendors are deployed. By using *Crypto-PAn*, we obtain in total 850 distinct IP addresses within 50 different/8 prefixes in this dataset. Similarly, the prefixes of these IP addresses are distributed non-uniformly.

To facilitate quantitative comparison between *PD-PAn* and *Multi-View*, we consider the number of injected IP addresses for each injected IP prefix is fixed in an IoT semantic attack. We use a pair of parameters (α, β) to specify an IoT semantic attack, where α denotes the proportion of injected IP prefixes and β denotes the number of injected IP addresses for each injected IP prefix. For example, $(\alpha = 0.1, \beta = 2)$ indicates that 10% of the IP prefixes are injected, and for each injected IP prefix, 2 distinct IP addresses are injected.

With these settings, we conduct experiments to evaluate *PD-PAn* on privacy, security and running time, and compare it with the state-of-the-art technique *Multi-View*. The experiments are performed on an edge node of an IoT environment, which is an Intel NUC Mini PC in Windows 10, with 16 GB memory and an Intel i7-1165G7 CPU @ 2.80 GHz. All compared methods are implemented in Java.

6.1. Evaluation on Privacy in Term of Anonymity

PD-PAn makes use of multiple fake views to conceal privacy patterns in the real view. We use the concept of k-anonymity to quantify privacy achieved by *PD-PAn*, and compare it with that achieved by *Multi-View*.

A real/fake view can be identified in any of the following ways. (1) Given two packets sent to/from the same injected IP prefix (i.e., one same injected IP address, or two injected IP addresses with the same prefix), if the anonymized IP prefixes in the two packets are different in a view, then this view is a fake view; (2) Given two different injected IP prefixes, if their anonymized forms are the same, then this view is a fake view; (3) Given several views, if the IP distribution in one view is different from that in other views, then this view is a real view.

PD-PAn achieves a better privacy level in comparison with *Multi-View* on both datasets as shown in Figure 4a. With the increase in the number of views v, the anonymity level achieved by *PD-PAn* grows linearly. If 50 views are generated, *PD-PAn* achieves around 40-anonymity among all the views (i.e., 10 fake views can be identified). As a comparison, *Multi-View* achieves hardly any privacy since the real view is always identifiable (recall Problem 3 introduced in Section 3.3). Figure 5a depicts the anonymity level on the number of views v with different settings on the prefix mask m. We consider three different masks: 0xff000000, 0xffff0000, and 0xffffff00. The anonymity level achieved by *PD-PAn* with all masks are close to each other. The proportion of indistinguishable fake views is always above 80%. Figure 6a depicts the anonymity level on the number of views v with different settings on the group size c. We consider five different grouping strategies, and let $c = 4, 8, 16, 32, 64$, respectively. The anonymity level achieved by *PD-PAn* with all considered group sizes is also close to each other. The proportion of indistinguishable fake views is slightly higher (i.e., the privacy is better) with a relatively small prefix group size.

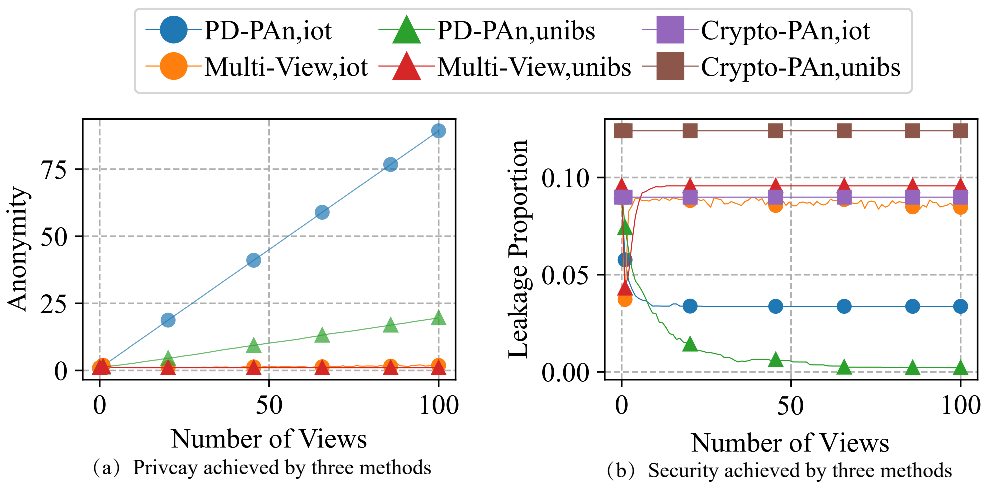

Figure 4. Privacy and security achieved by *PD-PAn*, *Multi-View*, and *Crypto-PAn* on `iot` dataset and `unibs` dataset (with m = 0xff000000, c = 32, α = 0.1, β = 1).

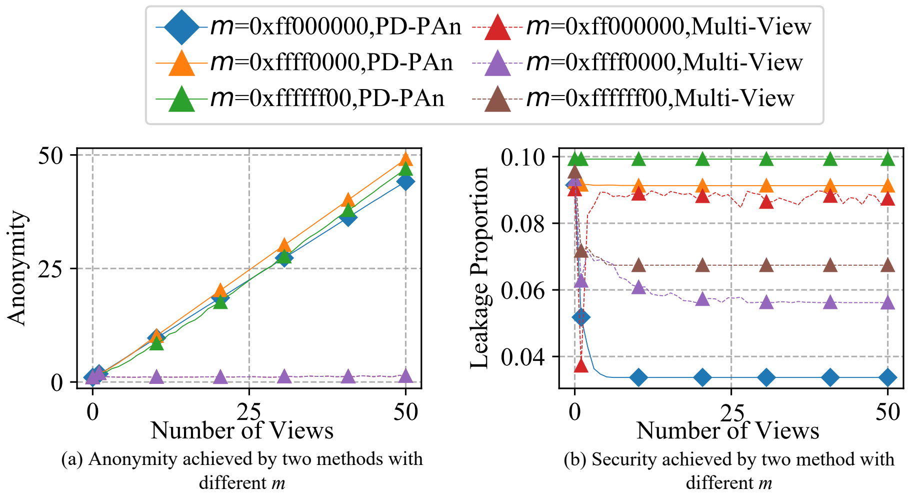

Figure 5. Anonymity and security achieved by *PD-PAN* and *Multi-View* with different prefix masks m (with c = 64, α = 0.1, β = 1).

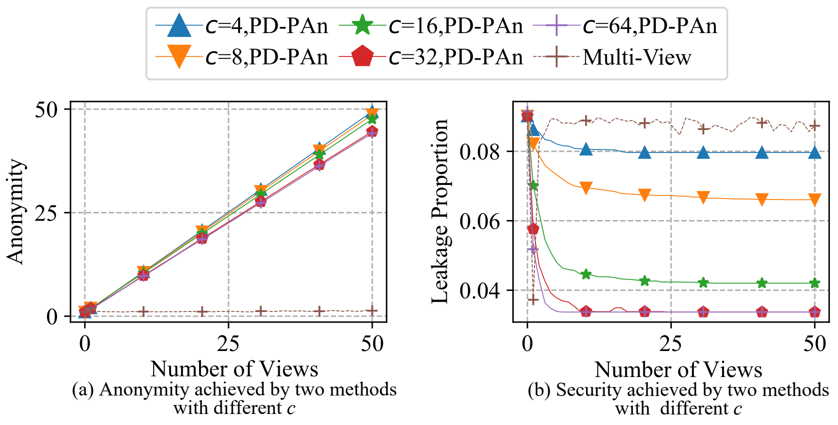

Figure 6. Anonymity and security achieved by *PD-PAN* and *Multi-View* with different prefix group size c (with m = 0xff000000, α = 0.1, β = 1).

We also evaluate anonymity achieved by *PD-PAn* and *Multi-View* under IoT semantic attacks. In all situations, *PD-PAn* performs much better than *Multi-View*. We consider the proportion of injected IP prefixes to be α, and the number of injected IP addresses for each injected IP prefix to be β. We first set $\beta = 1$, and consider α ranges in $[0.1, 0.5]$. The results are shown in Figure 7a. We then set $\alpha = 0.1$, and consider β ranges in $[1, 4]$. The results are shown in Figure 8a. The anonymity level decreases with α or β increases. When α reaches 0.5, which means 50% of IP prefixes are injected, *PD-PAn* cannot provide any anonymity.

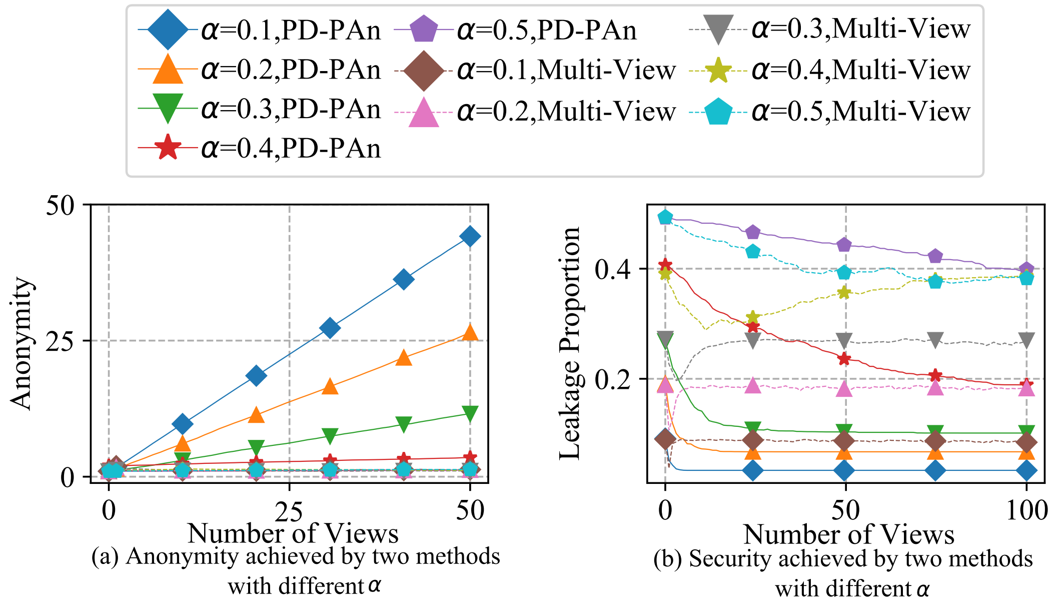

Figure 7. Anonymity and security achieved by *PD-PAn* and *Multi-View* with different values of α (with m = 0xff000000, c = 64, β = 1).

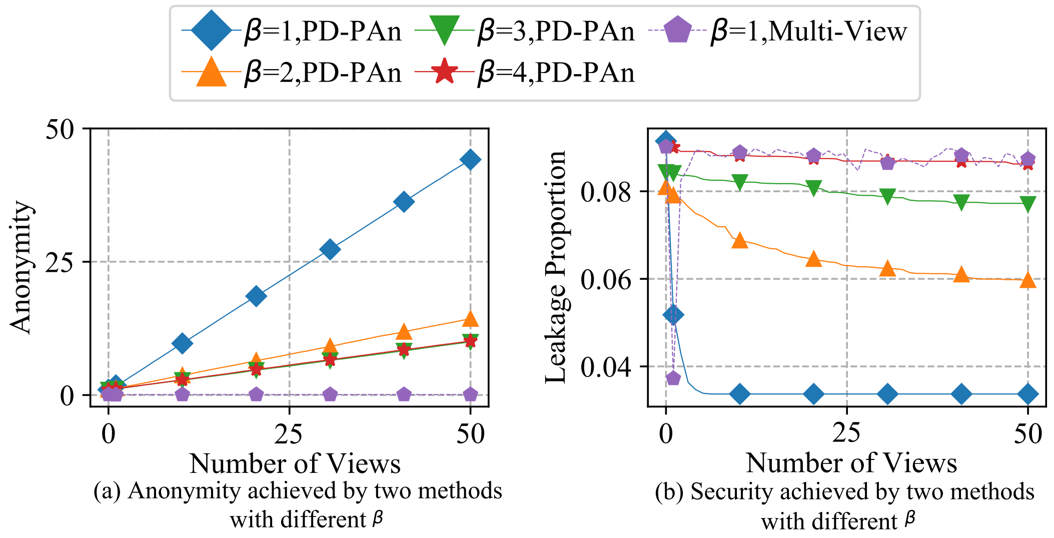

Figure 8. Anonymity and security achieved by *PD-PAn* and *Multi-View* with different values of β (with m = 0xff000000, c = 64, α = 0.1).

6.2. Evaluation on Secrecy in Term of Leaked Bits

We use the averaged proportion of leaked bits in an IP prefix to quantify secrecy achieved by *PD-PAn*, and compare it with that achieved by *Multi-View*. Under a semantic attack, some IP addresses are completely de-anonymized, and some IP addresses are partially de-anonymized. Consider two IP addresses, x and y. Their anonymized forms are x' and y', respectively. Further consider x is an injected IP address, so the adversary is able to obtain a mapping from x to x'. If x and y have a same prefix, then y' can be completely de-anonymized. Otherwise, x and y share k bits in common, then y' can be partially de-anonymized (k bits).

PD-PAn achieves a better secrecy level in comparison with *Multi-View* on both datasets as shown in Figure 4b. With the increase in the number of views v, the averaged proportion of leaked bits in *PD-PAn* first decreases rapidly (i.e., the secrecy level increases), then slows down after generating 10 to 20 views, and eventually approximates a value. Figure 5b depicts the secrecy level on the number of views v with different settings on the prefix mask m. We again consider three different masks 0xff000000, 0xffff0000, and 0xffffff00. For each mask considered, *PD-PAn* achieves a higher secrecy level than *Multi-View*. There is a difference in the secrecy levels achieved by *PD-PAn* with different masks. The fewer bits considered as prefixes, the more secrecy is better. Figure 6b depicts the secrecy level on the number of views v with different settings on the group size c. We again consider five different grouping strategies, and let $c = 4, 8, 16, 32, 64$, respectively. With a larger group size, the secrecy achieved by *PD-PAn* is better. Since there are more prefixes in a larger group, it is easier to make these prefixes confusing with each other. Even when the group size is extremely small ($c = 2$), *PD-PAn* achieves a better secrecy level than *Multi-View*.

We also evaluate secrecy achieved by *PD-PAn* and *Multi-View* under IoT semantic attacks. We first set $\beta = 1$, and consider α ranges in $[0.1, 0.5]$. The results are shown in Figure 7b. We then set $\alpha = 0.1$, and consider β ranges in $[1, 4]$. The results are shown in Figure 8b. In the overall trend, the averaged proportion of leaked bits decreases with the number of views increases, and finally approximates a convergence value. In all situations, *PD-PAn* performs better and more steadily than *Multi-View*.

6.3. Evaluation on Running Time

We compare the running time of *PD-PAn* and that of *Multi-View*. We set the number of views $v = 200$, and the prefix mask $m =$ 0xff000000, 0xffff0000 or 0xffffff00. In addition, we set the group size $c = 2, 4, 8, 16, 32$ for *PD-PAn*. The results are as shown in Table 3.

Table 3. Running time (seconds).

Method	$m =$ 0xff000000	$m =$ 0xffff0000	$m =$ 0xffffff00
PD-PAn (c = 2)	0.731	1.102	117.82
PD-PAn (c = 4)	0.710	1.096	57.911
PD-PAn (c = 8)	0.757	1.102	52.785
PD-PAn (c = 16)	0.831	1.147	50.894
PD-PAn (c = 32)	1.043	1.194	51.459
Multi-View	44.253	80.813	66.191

In most situations, *PD-PAn* is performed much faster than *Multi-View*, especially when the prefix domain is relatively small (when $m =$ 0xff000000 or 0xffff0000). The running time required by *PD-PAn* is only about 1/50 of that by *Multi-View*. However, when the prefix domain is large ($m =$ 0xffffff00), the running time of *PD-PAn* increases greatly. If the group size is very small ($c = 2$), there will be two many prefix groups. Due to this reason, *PD-PAn* requires an indeed large memory space to manage the grouping structure for all the IP prefixes. Without an additional user-space memory management policy, the IP prefixes will be frequently paged in/out, and thus, the whole process starts thrashing.

7. Related Work

Traffic anonymization enables organizations to preserve privacy in outsourcing NTA tasks. In this section, we briefly survey the traffic anonymization techniques, the de-anonymization attacks and existing defenses.

7.1. Network Traffic Anonymization

The problem of prefix-preserving IP anonymization has been widely studied for decades. Among the existing techniques, there are two most important algorithms: TCPDpriv [28], which was proposed in 1996, and *Crypto-PAn* [5], which was proposed in

2004. Other prefix-preserving NTA tools and services are mainly based on either of these two algorithms. TCPDpriv [28], proposed by Greg, uses random mapping tables to achieve anonymity. This algorithm is adopted in some NTA tools including IPSumDump [29], Zeek (formerly Bro) [30] and TCPDump Anonymizer [31], etc. *Crypto-PAn* [5] proposed by Fan et al. is a prefix-preserving IP anonymization algorithms based on cryptography. It has been deployed, implemented, or extended by a wide range of NTA tools and services including CoralReef [32], FLAIM [33], NFDump [34], TCPmkpub [35], PktAnon [36], Libtrace [37], etc.

7.2. Defenses against De-Anonymization Attacks

Researchers have proposed various de-anonymization attacks. Most of these attacks are based on traffic injection [38,39], fingerprinting [40,41] and crypt-analysis [5]. Attacks based on crypt-analysis can be defended against by employing new cryptography-based schemes. A semantic attack is a combination of traffic injection and fingerprinting. It has already been verified that any prefix-preserving IP anonymization algorithm is vulnerable to this attack since an adversary is able to make use of preserved prefixes to de-anonymize IP addresses [6,7]. Due to this reason, most existing defenses against this attack suffer from a drawback in either the effectiveness or functionality of NTA. A recent approach [8,42] uses multiple views of traffic logs to defend against semantic attacks. The data utility is preserved in the real view (but the prefix relationship is not preserved), and the privacy is preserved since the real view and fake views are indistinguishable. It has already been adopted in some research [43–46]. However, this technique has its limitations, as discussed in Section 3.3. The proposed method named PD-PAn can address these issues, which is the first traffic anonymization method that is both prefix-preserving and resistant to semantic attacks.

8. Conclusions

We investigated the semantic attacks in IoT scenarios. In contrast with the active, covert, and more dangerous IoT semantic attacks, we show that the state-of-the-art defense against semantic attacks suffers from significant drawbacks. We proposed *PD-PAn*, a prefix- and distribution-preserving traffic anonymization method, which ensures both data utility and privacy of network traffic logs. We also conducted intensive experiments on a public dataset and a private real-world dataset. The results validate the advancements of *PD-PAn*. However, *PD-PAn* requires large memory space to manage the grouping structure for all the IP prefixes. We leave the algorithm optimization for *PD-PAn* as future work.

Author Contributions: Conceptualization, K.D.; Methodology, X.G.; Validation, X.G.; Formal analysis, K.D.; Writing—original draft, X.G.; Writing—review & editing, K.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the Natural Science Foundation of China (Nos. 62102084 and 62072103), Jiangsu Provincial Natural Science Foundation of China (No. BK20190340), Jiangsu Provincial Key R&D Program (Nos. BE2021729. BE2022680, and BE2022065-4), Jiangsu Provincial Key Laboratory of Network and Information Security (No. BM2003201), and Key Laboratory of Computer Network and Information Integration of Ministry of Education of China (No. 93K-9).

Data Availability Statement: A publicly available dataset is analyzed in this study. This data can be found here: [http://netweb.ing.unibs.it/~ntw/tools/traces/ (accessed on 16 October 2023)].

Conflicts of Interest: The authors declare no conflict of interest. And the funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
NTA	Network Traffic Analysis
PP	Prefix-Preserving
RPP	Reverse Prefix-Preserving
CDF	Cumulative Density Function

References

1. Gartner. Market Guide for Network Traffic Analysis. Available online: https://www.gartner.com/en/documents/3902353 (accessed on 15 July 2022).
2. Oltsik, J. ESG White Paper: Network Traffic Analysis (NTA): A Cybersecurity 'Quick Win'. 2020. Available online: https://www.readkong.com/page/network-traffic-analysis-nta-a-cybersecurity-quick-win-5804677 (accessed on 15 July 2022).
3. AuvikNetworks. Auvik. Available online: https://www.auvik.com/ (accessed on 15 July 2022).
4. MixMode. Available online: https://mixmode.ai/ (accessed on 15 July 2022).
5. Fan, J.; Xu, J.; Ammar, M.H.; Moon, S.B. Prefix-preserving IP address anonymization: Measurement-based security evaluation and a new cryptography-based scheme. *Comput. Netw.* **2004**, *46*, 253–272. [CrossRef]
6. Brekne, T.; Årnes, A. Circumventing IP-address pseudonymization. In *Proceedings of the Communications and Computer Networks (NetCom)*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 43–48.
7. King, J.; Lakkaraju, K.; Slagell, A. A taxonomy and adversarial model for attacks against network log anonymization. In Proceedings of the ACM Symposium on Applied Computing (SIGAPP), Honolulu, HI, USA, 8–12 March 2009; pp. 1286–1293.
8. Mohammady, M.; Wang, L.; Hong, Y.; Louafi, H.; Pourzandi, M.; Debbabi, M. Preserving both privacy and utility in network trace anonymization. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS), Toronto, ON, Canada, 15–19 October 2018; pp. 459–474.
9. Bovenzi, G.; Aceto, G.; Ciuonzo, D.; Montieri, A.; Persico, V.; Pescapé, A. Network anomaly detection methods in IoT environments via deep learning: A Fair comparison of performance and robustness. *Comput. Secur.* **2023**, *128*, 103167. [CrossRef]
10. Mirsky, Y.; Doitshman, T.; Elovici, Y.; Shabtai, A. Kitsune: An ensemble of autoencoders for online network intrusion detection. *arXiv* **2018**, arXiv:1802.09089.
11. Bovenzi, G.; Aceto, G.; Ciuonzo, D.; Persico, V.; Pescapé, A. A hierarchical hybrid intrusion detection approach in IoT scenarios. In Proceedings of the GLOBECOM 2020–2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–7.
12. Zhang, W.; Meng, Y.; Liu, Y.; Zhang, X.; Zhang, Y.; Zhu, H. HoMonit: Monitoring smart home apps from encrypted traffic. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS), Toronto, ON, Canada, 15–19 October 2018; pp. 1074–1088.
13. Trimananda, R.; Varmarken, J.; Markopoulou, A.; Demsky, B. Packet-level signatures for smart home devices. In Proceedings of the The Network and Distributed System Security (NDSS) Symposium, San Diego, CA, USA, 23–26 February 2020.
14. Kai, D.; Yakun, Z.; Yuchen, Z.; Daoming, L.; Zhen, L.; Wenjia, W.; Xiaorui, Z. Real-Time Execution of Trigger-Action Connection for Home Internet-of-Things. In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), Virtual Conference, 2–5 May 2022.
15. Durstenfeld, R. Algorithm 235: Random permutation. *Commun. ACM* **1964**, *7*, 420. [CrossRef]
16. Brekne, T.; Årnes, A.; Øslebø, A. Anonymization of ip traffic monitoring data: Attacks on two prefix-preserving anonymization schemes and some proposed remedies. In Proceedings of the International Workshop on Privacy Enhancing Technologies (PETS), Cavtat, Croatia, 30 May–1 June 2005; pp. 179–196.
17. Yen, T.F.; Huang, X.; Monrose, F.; Reiter, M.K. Browser fingerprinting from coarse traffic summaries: Techniques and implications. In Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), Milan, Italy, 9–10 July 2009; pp. 157–175.
18. OConnor, T.; Mohamed, R.; Miettinen, M.; Enck, W.; Reaves, B.; Sadeghi, A.R. HomeSnitch: Behavior Transparency and Control for Smart Home IoT Devices. In Proceedings of the ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec), Miami, FL, USA, 15–17 May 2019; pp. 128–138.
19. Möllers, F.; Seitz, S.; Hellmann, A.; Sorge, C. Short paper: Extrapolation and prediction of user behaviour from wireless home automation communication. In Proceedings of the ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec), Oxford, UK, 23–25 July 2014; pp. 195–200.
20. Copos, B.; Levitt, K.; Bishop, M.; Rowe, J. Is anybody home? Inferring activity from smart home network traffic. In Proceedings of the IEEE Security and Privacy (SP) Workshops, San Jose, CA, USA, 22–26 May 2016; pp. 245–251.
21. Acar, A.; Fereidooni, H.; Abera, T.; Sikder, A.K.; Miettinen, M.; Aksu, H.; Conti, M.; Sadeghi, A.R.; Uluagac, S. Peek-A-Boo: I see your smart home activities, even encrypted! In Proceedings of the ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec), San Antonio, TX, USA, 16–19 May 2022; pp. 207–218.

22. Charyyev, B.; Gunes, M.H. IoT event classification based on network traffic. In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM) Workshops, Toronto, ON, Canada, 6–9 July 2020; pp. 854–859.
23. Wan, Y.; Xu, K.; Xue, G.; Wang, F. Iotargos: A multi-layer security monitoring system for internet-of-things in smart homes. In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), Toronto, ON, Canada, 6–9 July 2020; pp. 874–883.
24. Aceto, G.; Ciuonzo, D.; Montieri, A.; Pescapé, A. Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges. *IEEE Trans. Netw. Serv. Manag. (TNSM)* **2019**, *16*, 445–458. [CrossRef]
25. Liu, Y.; Gu, R.; Yang, Z.; Ji, Y. Hierarchical community discovery for multi-stage IP bearer network upgradation. *J. Netw. Comput. Appl. (JNCA)* **2021**, *189*, 103151. [CrossRef]
26. Dusi, M.; Crotti, M.; Gringoli, F.; Salgarelli, L. Detection of encrypted tunnels across network boundaries. In Proceedings of the IEEE International Conference on Communications (ICC), Crete, Greece, 6–8 August 2008; pp. 1738–1744.
27. Este, A.; Gringoli, F.; Salgarelli, L. On-line SVM traffic classification. In Proceedings of the IEEE International Conference on Wireless Communications and Mobile Computing (WCMC), Istanbul, Turkey, 5–8 July 2011; pp. 1778–1783.
28. Greg, M. Tcpdpriv Release 1.1.11. Available online: https://fly.isti.cnr.it/software/tcpdpriv/ (accessed on 22 July 2022).
29. Kohler, E. IPSumDump Version 1.86. Available online: https://altlinux.pkgs.org/p10/classic-aarch64/ipsumdump-1.86-alt3.aarch64.rpm.html (accessed on 22 July 2022).
30. Zeek. The Zeek Network Security Monitor. Available online: https://zeek.org/ (accessed on 22 July 2022).
31. Claudio, M. Tcpdump Anonymizer. Available online: https://sourceforge.net/projects/anonymizer/ (accessed on 22 July 2022).
32. CAIDA. CoralReef Software Suite. Available online: https://www.caida.org/catalog/software/coralreef/ (accessed on 22 July 2022).
33. Slagell, A.J.; Lakkaraju, K.; Luo, K. FLAIM: A multi-level anonymization framework for computer and network logs. In Proceedings of the Large Installation System Administration (LISA) Conference, Washington, DC, USA, 3–8 December 2006; Volume 6, pp. 3–8.
34. Haag, P. NFDump Release v1.6.24. Available online: https://gitee.com/mirrors_phaag/nfdump/tags (accessed on 22 July 2022).
35. LBNL; ICSI. TCPmkpub Version 0.1. Available online: https://www.icir.org/enterprise-tracing/tcpmkpub.html (accessed on 22 July 2022).
36. Christoph, P.M.; Thomas Gamer, M.S. PktAnon—Packet Trace ANONYMIZATION, Version 1.4.0-dev. Available online: https://www.tm.kit.edu/software/pktanon/index.html (accessed on 22 July 2022).
37. Alcock, S.; Lorier, P.; Nelson, R. Libtrace. Available online: https://github.com/LibtraceTeam/libtrace (accessed on 22 July 2022).
38. Bethencourt, J.; Franklin, J.; Vernon, M.K. Mapping internet sensors with probe response attacks. In Proceedings of the USENIX Security Symposium, Baltimore, MD, USA, 31 July–5 August 2005; pp. 193–208.
39. Pang, R.; Allman, M.; Paxson, V.; Lee, J. The devil and packet trace anonymization. *ACM SIGCOMM Comput. Commun. Rev. (CCR)* **2006**, *36*, 29–38. [CrossRef]
40. Kohno, T.; Broido, A.; Claffy, K.C. Remote physical device fingerprinting. *IEEE Trans. Dependable Secur. Comput. (TDSC)* **2005**, *2*, 93–108. [CrossRef]
41. Coull, S.E.; Wright, C.V.; Monrose, F.; Collins, M.P.; Reiter, M.K. Playing devil's advocate: Inferring sensitive information from anonymized network traces. In Proceedings of the The Network and Distributed System Security (NDSS) Symposium, San Diego, CA, USA, 28 February–2 March 2007; Volume 7, pp. 35–47.
42. Mohammady, M.; Oqaily, M.; Wang, L.; Hong, Y.; Louafi, H.; Pourzandi, M.; Debbabi, M. A Multi-view approach to preserve privacy and utility in network Trace ANONYMIZATION. *ACM Trans. Priv. Secur. (TOPS)* **2021**, *24*, 1–36. [CrossRef]
43. Lim, H.W.; Poh, G.S.; Xu, J.; Chittawar, V. PrivateLink: Privacy-Preserving Integration and Sharing of Datasets. *IEEE Trans. Inf. Forensics Secur. (TDSC)* **2019**, *15*, 564–577. [CrossRef]
44. Xie, S.; Wang, H.; Wang, S.; Lu, H.; Hong, Y.; Jin, D.; Liu, Q. Homogeneous and Mixed Energy Communities Discovery with Spatial-Temporal Net Energy. *arXiv* **2019**, arXiv:1902.03916.
45. Bienias, P.; Warzyński, A.; Kołaczek, G. Application and preliminary evaluation of Anontool applied in the anomaly detection module. In Proceedings of the IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), Paris, France, 10–12 June 2020; pp. 119–123.
46. Xiong, Y.; Ramachandran, G.K.; Ganesan, R.; Jajodia, S.; Subrahmanian, V. Generating Realistic Fake Equations in Order to Reduce Intellectual Property Theft. *IEEE Trans. Dependable Secur. Comput. (TDSC)* **2020**, *19*, 1434–1445. [CrossRef]