Correlating Network Events and Transferring Labels in the Presence of IP Address Anonymisation

Sebastian Abt[†,‡] and Harald Baier[†]
[†]da/sec – Biometrics and Internet Security Research Group, http://www.dasec.h-da.de
[‡]DE-CIX Research and Development, http://www.de-cix.net
sebastian.abt@de-cix.net, harald.baier@h-da.de

***Abstract*—The availability of labelled data, i.e. ground-truth or reference data, is typically a requirement for performing network research, especially for network security research. Labelled data, however, are sparsely available. Data sets present in repositories such as CAIDA or PREDICT are mostly missing labels and have IP addresses anonymised. Especially the latter compounds correlating these data sets with third-party information in order to assign labels a posteriori. To address this problem, we propose a scheme to anonymise IP addresses such that later correlation is still possible, without compromising security of either data sponsoring entity. The scheme we propose is based on Crypto-PAn [1] and is able to correlate events using anonymised IP addresses as correlation keys, without restricting choice of the cryptographic secret.**

I. Introduction

Obtaining labelled real-world network traffic samples is crucial for network research. However, sharing network traffic is constrained by legal requirements and often seen critical by data owners [2]. If an entity is able to share traces from its network, anonymisation is typically applied. In past, different anonymisation techniques have been proposed by researchers (e.g., [1], [3], [4]). Two frequently applied methods are removal of payload data and anonymisation of IP addresses. Given a plethora of known and unknown protocols carried as payload in IP packets, the first is clearly required in order to not accidentally reveal any personally-identifiable information (PII) hidden in the payload (e.g., email addresses, usernames, passwords, etc.) and to guarantee secrecy and privacy of correspondence. While still open to question if IP addresses have to be regarded as PII, these addresses are typically anonymised as matter of precaution and, amongst other things, to not reveal information about possibly vulnerable devices or about internal network topology [5]. In general, while agreeing to share, a data owner at the same time typically tries hard to not reveal any information that could be used against him, in whatever case. Obviously, these two goals tend to contradict and, as consequence, the resulting data may not be usable universally, but only for specific tasks.

One illustrating example of this phenomenon are the CAIDA Anonymized Internet Traces[1] data sets. These data sets provide passively captured yearly network traffic samples as seen on high-speed backbone links since 2008. Hence, a valuable source of information that represents an Internet-scale view on network traffic and can be used to study changes over time. However, these traces provide no payload and IP addresses are anonymised using Crypto-PAn [1], the de facto standard for IP address anonymisation. What is even more of a problem: these traces come without labels. Hence, the traces provide no information about events that may be reflected in these data sets. And as traces are anonymised, correlation of events with other third-party information, such as blacklists (e.g., [6], [7]) or results from honeypots (e.g., [8], [9]) and darknets (e.g., [10], [11]), is at least difficult, but mostly impossible. The availability of such labelled data, commonly also referred to as ground-truth or reference data, however, is of utmost importance for network research – either for a human to understand the data at hand or for a machine to deduce classification models [12]. Hence, utility of the CAIDA Anonymized Internet Traces is unfortunately limited for a specific range of applications[2].

Especially the ability to annotate, i.e. assign labels to, a given anonymised data set a posteriori, i.e. after it has been collected, anonymised and published, would greatly benefit the research community as it would allow the information and, hence, value of a data set to grow over time. In this paper, we propose a scheme which allows exactly that. The method we describe here enables a data repository to correlate anonymised traffic samples of one data owner with events present in third-party information of another data owner. Consequently, our scheme allows to transfer labels from one data set to another one, even in the presence of IP prefix anonymisation.

The remainder of this paper is structured as follows: In Section II we briefly recapitulate relevant background on prefix-preserving IP address anonymisation and especially Crypto-PAn [1], which our scheme bases on. In Section III we describe our scheme and in Section IV we discuss design aspects, security properties and practical considerations. In Section V we present related work and we conclude in Section VI.

II. Background

A. Prefix-preserving IP Address Anonymisation

Computer networks are made up of interconnected segments of hosts. A segment typically shares a specific IP subnet, which

[1] https://www.caida.org/data/passive/passive_trace_statistics.xml

[2] The same conclusion applies to other unlabelled and anonymised data sets. This CAIDA data set was simply meant to give an example and we especially appreciate the existence of this data set and the efforts of the data owners.

978-3-901882-85-2

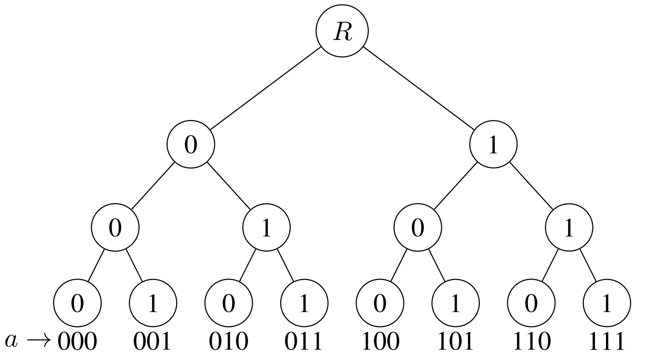


Figure 1. Illustration of an address tree with $n = 3$: R denotes root, $0, 1$ denote address bit at specific position (cf. to [1]).

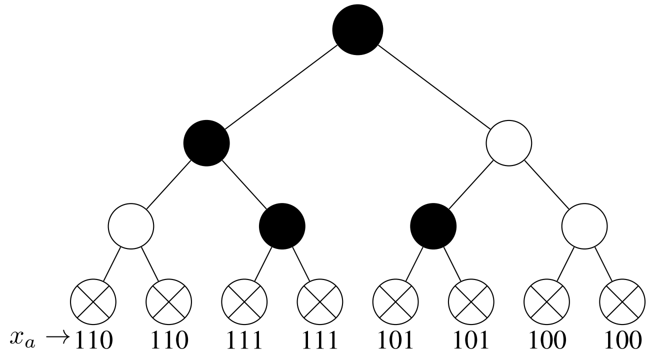


Figure 2. Illustration of an anonymisation tree with $n = 3$: ● denotes address bit negation, ○ denotes address bit identity, $\otimes$ denotes leaf (cf. to [1]).

is used to enumerate the hosts within the segment. A subnet is of a specific bit length, say m, and hence allows to assign unique IP addresses to at most $2^m - 2$ hosts in that subnet[3]. The remaining $n - m$ bits of an IP address of length n are commonly referred to as the prefix of the subnet and are used in global IP routing to define routing paths to the hosts within a subnet.

Prefix-preservation is an important property for being able to use anonymised data for specific research questions. For instance, detection of events spanning across a whole subnet (e.g., vulnerability scanning) is only possible if anonymised IP addresses preserve the prefix information present in the original IP addresses. If the prefix information is lost during anonymisation, the concept of subnets and, hence, the correspondence to a specific subnet, is lost as well, by definition. Formally, prefix-preservation can be defined as follows: let $a = a_1 a_2 a_3 \cdots a_n$ and $b = b_1 b_2 b_3 \cdots b_n$ denote two IP addresses of length n bit ($n \in \mathbb{N}$) (e.g., $n = 32$ for IPv4 addresses and $n = 128$ for IPv6) and $a' = a'_1 a'_2 a'_3 \cdots a'_n = F(a), b' = b'_1 b'_2 b'_3 \cdots b'_n = F(b)$ denote two anonymised IP addresses computed using an anonymisation function $F(\cdot)$.

Definition 1 (Prefix-Preservation): An anonymisation function $F(\cdot)$ is prefix-preserving, if, and only if, for any two IP addresses a, b: $a'_i = b'_i \Leftrightarrow a_i = b_i$ for all $i = 1, \ldots, k \leq n, (i, k \in \mathbb{N})$.

More specifically, we say that two IP addresses a, b share a k-bit prefix if, and only if, $a_1 a_2 \cdots a_k = b_1 b_2 \cdots b_k \wedge a_{k+1} \neq b_{k+1}$. Consequently, in that case, a prefix-preserving anonymisation function $F(\cdot)$ would compute a', b' such that $a'_1 a'_2 \cdots a'_k = b'_1 b'_2 \cdots b'_k \wedge a'_{k+1} \neq b'_{k+1}$.

It is important to note at this point that we do *not* require $a_1 a_2 \cdots a_k = b_1 b_2 \cdots b_k = a'_1 a'_2 \cdots a'_k = b'_1 b'_2 \cdots b'_k$, i.e. we do *not* require that the actual prefix is preserved during anonymisation. However, we require that the relevant prefix information, i.e. the length k, of the prefix is preserved after applying $F(\cdot)$.

The definition above shows that our anonymisation function $F(\cdot)$ is of the form $F : \mathbb{B}^n \rightarrow \mathbb{B}^n$. Moreover, it can be shown that every prefix-preserving anonymisation function $F(\cdot)$ is a bijective function (for proof, please refer to [1]) and, hence, a permutation of the domain. While this limitation in the co-domain affects security as entropy is limited to n bit, this choice is reasonable from a practical point of view: specific data representations (e.g., pcap formatting) expect IP addresses, or its corresponding anonymised representations, to fit into 4 (IPv4) or 16 (IPv6) bytes of memory.

In [1] the authors propose a tree-based visual representation of prefix-preserving anonymisation as illustrated in Figures 1 and 2. Figure 1 illustrates an IP address space. For brevity, this address space is limited to length $n = 3$ bit. The address space totals 2^n addresses ranging from 0 to $2^n - 1$. In Figure 2 we show one possible anonymisation tree as example that can be used to anonymise, i.e. permute, addresses of the address space. More specifically, Figures 1 and 2 visualise the canonical form theorem presented in [1], which states that every prefix-preserving anonymisation function $F(\cdot)$ takes the form

$$F(a) = (a_1 \oplus f_0) \cdots (a_n \oplus f_{n-1}(a_1, \cdots, a_{n-1})), \quad (1)$$

where $(a_i \oplus f_{i-1}(a_1, \cdots, a_{i-1})) = a'_i$ $(i = 1, \ldots, n)$ and f_0 being constant[4]. This is equivalent to writing

$$F(a) = (a_1 \cdots a_n) \oplus x_a = a \oplus x_a, \quad (2)$$

with

$$x_a = (f_0 \cdots f_{n-1}(a_1, \cdots, a_{n-1})). \quad (3)$$

Functions $f_i : \mathbb{B}^i \rightarrow \mathbb{B}$ can be obtained from the anonymisation tree presented in Figure 2 by following the path from root to leaf nodes, where a filled circle denotes bit negation, i.e. $f_i(a_1, \ldots, a_i) = 1$ and an empty circle denotes bit identity, i.e. $f_i(a_1, \ldots, a_i) = 0$. Hence, permutation vectors x_a are bit strings of length n, derived by concatenating binary digits corresponding to the specific vertices in the path from root to the leaf given by address a.

What can be seen from the definition and is nicely visualised in the tree representation given in Figure 2 is that the maximum number of permutation vectors x_a induced by the

[3] 2^m available addresses minus two addresses reserved for network and broadcast.

[4] Here, and throughout the remainder of this work, $\oplus$ denotes exclusive-or.

anonymisation tree is 2^{n-1} as the bit value corresponding to the root vertex, i.e. f_0, is constant. This means that at least two different IP addresses a, b share the same permutation vectors, i.e. $x_a = x_b$. More specifically, that is exactly the case if a, b share an $n-1$-bit prefix and, hence, differ only in the least significant bit. However, depending on the anonymisation tree used, there may be more distinct pairings of IP addresses sharing the same permutation vectors. The lower bound of the number of different permutation vectors is 1, which is exactly the case if $f_i(a_1, \ldots, a_i) = f_j(a_1, \ldots, a_j)$ for $i, j \geq 1$, i.e. if the permutation vector is essentially defined by choice of f_0.

B. Cryptography-based Prefix-preserving Anonymisation

Naturally, the question arises how to choose a good anonymisation tree and how to induce such tree efficiently. For that purpose, Xu et al. [1] propose cryptography-based prefix-preserving anonymisation, briefly commonly referred to as Crypto-PAn. This method uses cryptographic ciphers to induce an anonymisation tree. In Crypto-PAn, functions f_i are defined as follows: $f_i(a_1, \cdots, a_i) := L(R(P(a_1 \cdots a_i), \kappa))$, with $L(\cdot)$ denoting the function that extracts the least significant bit of a given bit string, i.e. a substring of length 1; $R(\cdot)$ denoting a cryptographically-strong pseudo-random permutation function; and $P(\cdot)$ denoting a padding function that extends the bit string $a_1 \cdots a_i$ of length i to a bit string of the length required by the pseudo-random permutation function $R(\cdot)$.

The implementation[5] provided by Xu et al. uses the Rijndael [13] block cipher with a block length of 128 bit in order to derive pseudo-random permutations and relies on a 256 bit secret $\kappa = \kappa_0\kappa_1 \cdots \kappa_{255}$. This secret key is split into two 128 bit strings $\kappa' = \kappa_0 \cdots \kappa_{127}$ and $\kappa'' = \kappa_{128} \cdots \kappa_{255}$. κ' is used as secret key of the pseudo-random permutation function and κ'', after being initially encrypted, is used as pad to extend addresses of length $n < 128$ bit to the required block length.

In case of Crypto-PAn, an anonymisation tree is induced using Rijndael and deterministically defined by a single 256 bit secret κ. This means, using the same secret κ will lead to exactly the same anonymisation tree. As Rijndael is a cryptographic block cipher, one can assume that $Prob[f_i(a_1, \cdots, a_i) = 0] = Prob[f_i(a_1, \cdots, a_i) = 1] = 0.5$, i.e. that the probability of $f_i(a_1, \cdots, a_i)$ taking 0 is the same of it taking 1. Hence, the numbers of filled and empty nodes of the anonymisation tree will be close to each other and permutation vectors x_a will likely differ. More specifically, it is safe to assume that the case $f_i(a_1, \cdots, a_i) = f_j(a_1, \cdots, a_j)$ for $i, j > 1$ is highly unlikely. In that sense, we regard Crypto-PAn as being able to induce good anonymisation trees.

III. Correlating Anonymised IP Addresses

So far, we especially recapitulated prefix-preserving IP address anonymisation as well as Crypto-PAn and studied specific important properties. Besides the aforementioned properties, what makes Crypto-PAn the effective de facto standard in prefix-preserving IP address anonymisation is that anonymisation can be performed at different sites and different time, while still preserving prefix length information across the data. Hence, Crypto-PAn facilitates distributed data collection and anonymisation. One prerequisite for this, however, is that the same secret κ is used at different sites.

While satisfying the latter requirement may be feasible for anonymisation across different sites of the same organisation (e.g. geographically diverse Points-of-Presence (PoPs) of big Internet Service Providers (ISP) or different establishments of the same company), this is inconceivable for sites belonging to different organisations. In that case, sharing κ would effectively de-anonymise anonymised IP addresses. Hence, if the same κ is used across different sites and it is not permitted to share real IP addresses between these sites, then anonymised IP addresses can neither be exchanged between these sites. In that case, anonymisation using the same secret κ at multiple sites is only useful if a third, highly trusted entity is correlating the data sets and will *not* release any data.

The latter scenario may be valid, but is not what we are intending. Our aim is to propose a scheme that allows to share data with anonymised IP addresses using different secrets κ_i at all sites while still preserving the prefix-length and, more specifically, being able to correlate IP addresses of one data set with events observed in other data sets. Most important, afterwards, we want to be able to publish the resulting correlated data set.

This section will introduce the scheme we propose on top of Crypto-PAn. In subsection III-A we briefly introduce the terminology used throughout the remainder of this paper and in subsection III-B we explain the trust model we assume for our scheme. In subsection III-C we elaborate our concept.

A. Terminology

Our concept relies on four distinct and collaborating entities which we describe as follows:

As *data owner* we refer to an entity that is legally and administratively responsible for a specific data set. Specifically, the data owner is the single entity that is able to collect a data set in its given constitution at a specific point in time. The data owner is aware of real IP addresses present within its data sets and required to anonymise IP addresses before being able to share data sets.

A *label provider* is a specific instance of data owner which is able to detect and annotate events within the data sets at its disposal.

The *data repository* is responsible for indexing and storing data sets received from data owners. Additionally, in our scheme, a data repository is capable of correlating anonymised IP addresses present in data sets received from data owners and contributing label providers and is able to transfer labels from the label provider data set to the data owner data set.

For our scheme, a *key distribution centre* is required and defined to be an entity that distributes relevant secrets and derived information between collaborating entities.

As an illustrative example, we can think of an ISP that is collecting and sharing data from within its network as a

[5] http://www.cc.gatech.edu/computing/Telecomm/projects/cryptopan/

data owner. A security researcher who is running honeypots or analysing darknet activity and deriving attacker patterns from these analyses may be one specific label provider. Both may be willing to collaborate and to choose a mutually trusted key distribution centre for obtaining secrets. Additionally, both submit their results to a mutually trusted data repository.

B. Trust Model

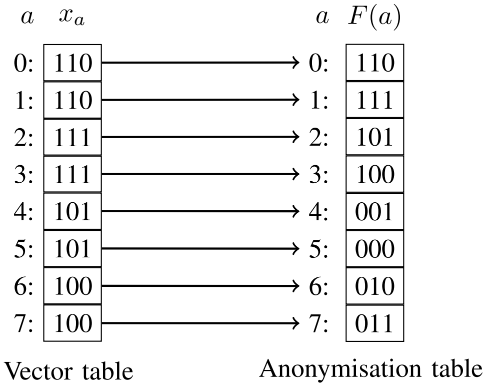
In the previous subsection we already mentioned *trust*, which is an important property for collaboration, especially if collaboration concerns highly sensitive data, such as network traces. For our scheme, we assume the following trust model:

(i) Data owner and label providers are willing to collaborate but do not trust each other. This is especially true from the data owners point of view. Hence, a data owner will never give a label provider access to traces containing real IP addresses, nor is a data owner willing to share its anonymisation secret κ with a label provider. However, a data owner is willing to give a label provider access to anonymised traces via a data repository.

(ii) We assume that both entities are able to identify a data repository which they both trust in the sense that they agree to submit anonymised data to the repository and that they agree the repository to obtain required information from a key distribution centre in order to being able to correlate events and transfer labels. However, data owner and label provider would not agree to sending anonymisation secrets to the repository.

(iii) Similarly, data owner and label provider can agree on a key distribution centre which they both rely on in order to collaborate. However, a data owner would never agree to using a secret provided by the key distribution centre for anonymisation of IP addresses present in its network traces.

We are convinced that the trust model we present above is valid and reasonable for the following reasons: First, the relationship between data owners and data repositories is already established in the wild. For instance, CAIDA, PREDICT[6] or DataCat[7] repositories store, index and publish data sets provided by data owners. The data sets found in these repositories have been anonymised by its owners and neither the true IP addresses, nor the anonymisation keys are known to the repository. As a label provider is a special instance of a data owner, this can be assumed valid for label providers as well. Second, we can observe in reality that data owners are willing to collaborate as data sets are already being shared mutually. In previous work [12] we performed a study on the state-of-data in network security research and showed that researchers heavily utilise real-world data sets and from these 44% were provided by third-parties, i.e. data owners. Assuming that these data sets were provided only in anonymised form is likely to be true with high probability, given the sensitivity of the data and the legal framework. Third, we argue that data owners will be able to trust a key distribution centre, if they are not forced to use the information obtained from the key distribution centre as anonymisation secret or to deterministically derive an anonymisation secret from this information. Put differently, agreeing on a key distribution centre that supports collaboration is feasible, if the key distribution centre is not able to break anonymisation after data sets have been published in the data repository using the information present at the key distribution centre.



a	x_a		a	$F(a)$
0:	110	→	0:	110
1:	110	→	1:	111
2:	111	→	2:	101
3:	111	→	3:	100
4:	101	→	4:	001
5:	101	→	5:	000
6:	100	→	6:	010
7:	100	→	7:	011

Figure 3. Table representation of the anonymisation process.

C. Our Scheme

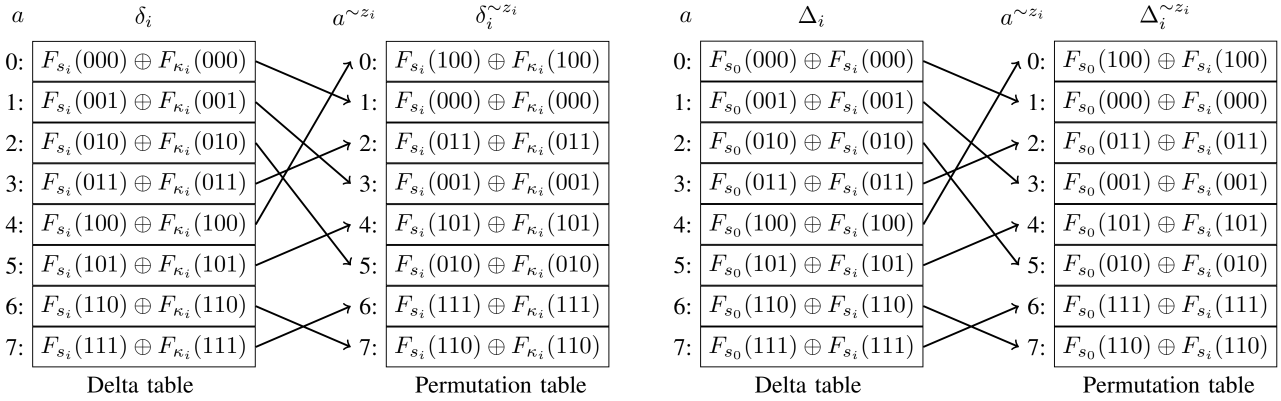
The scheme we propose can be divided into two different phases: during the bootstrap sequence, parameters required to correlate anonymised IP addresses are distributed by the key distribution centre and all tables and matrices are set up. Afterwards, data owner and label provider can send data to the data repository where correlation and transfer of labels happen. In what follows, we are going to discuss these two phases in more detail. Afterwards, we formally prove the correctness of our scheme.

Before however, we would like to introduce a different point of view on prefix-preserving anonymisation than already given in Section II-A. The previous interpretation was given using anonymisation trees, which is based on Xu et al. [1]. As an extension to this representation, we already introduced the notion of permutation vectors x_a in equation (3) which we use to derive an anonymised representation $F_\kappa(a)$ of IP address a by computing $F_\kappa(a) = a \oplus x_a$. From this notion, a table representation of prefix-preserving IP address anonymisation naturally emerges. As example, a table representation corresponding to the trees provided in Figures 1 and 2 with $n = 3$ is illustrated in Figure 3. Please note that for the remainder we will parametrise the anonymisation function $F(\cdot)$ introduced in section II with a subscript in order to explicitly denote which secret has been used to induce the anonymisation tree.

In our table representation an IP address a can be regarded as index to a table holding the permutation vector x_a corresponding to a. Obviously, there is an one-to-one mapping between the table representation we introduce and the tree representation introduced by Xu et al.: the table at index a contains the concatenation of binary digits corresponding to the nodes of an anonymisation tree on the path from root

[6] https://www.predict.org/

[7] http://imdc.datcat.org/Home



a	δ_i	$a^{\sim z_i}$	$\delta_i^{\sim z_i}$
0:	$F_{s_i}(000) \oplus F_{\kappa_i}(000)$	0:	$F_{s_i}(100) \oplus F_{\kappa_i}(100)$
1:	$F_{s_i}(001) \oplus F_{\kappa_i}(001)$	1:	$F_{s_i}(000) \oplus F_{\kappa_i}(000)$
2:	$F_{s_i}(010) \oplus F_{\kappa_i}(010)$	2:	$F_{s_i}(011) \oplus F_{\kappa_i}(011)$
3:	$F_{s_i}(011) \oplus F_{\kappa_i}(011)$	3:	$F_{s_i}(001) \oplus F_{\kappa_i}(001)$
4:	$F_{s_i}(100) \oplus F_{\kappa_i}(100)$	4:	$F_{s_i}(101) \oplus F_{\kappa_i}(101)$
5:	$F_{s_i}(101) \oplus F_{\kappa_i}(101)$	5:	$F_{s_i}(010) \oplus F_{\kappa_i}(010)$
6:	$F_{s_i}(110) \oplus F_{\kappa_i}(110)$	6:	$F_{s_i}(111) \oplus F_{\kappa_i}(111)$
7:	$F_{s_i}(111) \oplus F_{\kappa_i}(111)$	7:	$F_{s_i}(110) \oplus F_{\kappa_i}(110)$
	Delta table		Permutation table

(a) Delta table δ_i and corresponding permutation table $\delta_i^{\sim z_i}$.

a	Δ_i	$a^{\sim z_i}$	$\Delta_i^{\sim z_i}$
0:	$F_{s_0}(000) \oplus F_{s_i}(000)$	0:	$F_{s_0}(100) \oplus F_{s_i}(100)$
1:	$F_{s_0}(001) \oplus F_{s_i}(001)$	1:	$F_{s_0}(000) \oplus F_{s_i}(000)$
2:	$F_{s_0}(010) \oplus F_{s_i}(010)$	2:	$F_{s_0}(011) \oplus F_{s_i}(011)$
3:	$F_{s_0}(011) \oplus F_{s_i}(011)$	3:	$F_{s_0}(001) \oplus F_{s_i}(001)$
4:	$F_{s_0}(100) \oplus F_{s_i}(100)$	4:	$F_{s_0}(101) \oplus F_{s_i}(101)$
5:	$F_{s_0}(101) \oplus F_{s_i}(101)$	5:	$F_{s_0}(010) \oplus F_{s_i}(010)$
6:	$F_{s_0}(110) \oplus F_{s_i}(110)$	6:	$F_{s_0}(111) \oplus F_{s_i}(111)$
7:	$F_{s_0}(111) \oplus F_{s_i}(111)$	7:	$F_{s_0}(110) \oplus F_{s_i}(110)$
	Delta table		Permutation table

(b) Delta table Δ_i and corresponding permutation table $\Delta_i^{\sim z_i}$.

Figure 4. Illustration of delta tables and corresponding permutation tables for addresses of length $n = 3$. Permutation is defined by z_i.

to leaf of a. By applying the anonymisation function, i.e. computing $F_\kappa(a) = a \oplus x_a$, we obtain a similar table with $F_\kappa(a)$ at the position specified by index a. For the remainder, we will refer to the first table holding the permutation vectors as *vector table* and to the latter holding the anonymised IP addresses as *anonymisation table*.

1) Bootstrap Sequence: In order to be able to correlate events in presence of IP address anonymisation, our scheme relies on several parameters that have to be shared between the involved entities. This happens during the bootstrap sequence and is initiated by the key distribution centre. The key distribution centre issues bootstrap secrets s_0, s_i $(i \geq 1)$ to the data owner and label providers, respectively, as well as permutation secrets z_i which are shared between the data owner and the i-th label provider. Additionally, the key distribution centre computes the delta tables Δ_i bound to the data owner and the i-th label provider for all IP addresses $0 \leq a \leq 2^n - 1$. More formally, let $\Delta_{i,a}$ denote the a-th cell of table Δ_i, then $\Delta_{i,a} = F_{s_0}(a) \oplus F_{s_i}(a)$, i.e. the bit difference between IP address a anonymised using secret s_0 and a anonymised using secret s_i. Put differently, delta table Δ_i corresponds to the bit difference of the two anonymisation tables induced by s_0 and s_i. From delta table Δ_i a corresponding permutation table $\Delta_i^{\sim z_i}$ is derived. The permutation used here is defined by z_i, i.e. the secret shared between the data owner and the i-th label provider, and is applied to the table index a, as illustrated in Figure 4b. The permutation table $\Delta_i^{\sim z_i}$ is issued to the data repository for data correlation. With superscript $\sim z_i$ we denote this permutation throughout the remainder.

In a second step, the data owner as well as the label providers each compute private secrets κ_0, κ_i $(i \geq 1)$, respectively. κ_i $(i \geq 0)$ are held secret and are not shared with any involved entity. All secrets s_i, κ_i, z_i $(i \geq 0)$ are of same length (e.g., 256 bit). Similarly to Δ_i, using s_i and κ_i the data owner as well as label providers compute delta tables δ_i for all IP addresses a as illustrated in Figure 4a. Let $\delta_{i,a}$ denote the a-th cell of table δ_i, then $\delta_{i,a} = F_{s_i}(a) \oplus F_{\kappa_i}(a)$, that is, the bit difference between anonymisation tables induced by the secrets s_i and κ_i. Afterwards, delta tables δ_i are permuted to obtain permutation tables $d_i^{\sim z_i}$ and permutation tables $\delta_i^{\sim z_i}$ are forwarded to the data repository.

This whole bootstrap sequence is illustrated in Figure 5. This illustration serves as an example with only one label provider, but can easily be extended to additional label providers. Specifically, data and label providers can be added incrementally over time without affecting existing collaborations. The parameters attached to the different entities using curly brackets summarise the information available to every entity after successful completion of the bootstrap sequence.

2) Event Correlation Phase: The event correlation phase requires two different steps: first, data owner and label providers have to anonymise IP addresses present in their data and need to exchange data with the data repository. Second, the data repository has to correlate events and transfer labels. Both steps are described in what follows.

a) IP Address Anonymisation and Data Exchange: After successful bootstrap, all entities have the required information in order to be able to anonymise IP addresses in network traces as well as to correlate events in traces of two collaborating entities. Anonymisation of IP addresses is straight forward: The data owner uses the secret κ_0 to initialise its specific implementation of Crypto-PAn [1] and anonymises IP addresses using this key or, more specifically, permutation vectors induced by this key. Formally, the data owner computes an anonymised IP address a' as follows: $a' = F_{\kappa_0}(a) = a \oplus x_a$, with x_a denoting the permutation vector found in the a-th cell of the vector table induced by κ_0. Afterwards, the data owner can send a set of anonymised network traces D to the data repository.

The label provider performs anonymisation of IP addresses exactly in the same way using its private secret κ_i instead and sends a set of event descriptors E_i, i.e. patterns describing events in network traffic, to the data repository as well. In addition to that, label provider i provides a sparse lookup table Π_i which contains a mapping of the cell indices of the anonymisation table used by the i-th label provider and

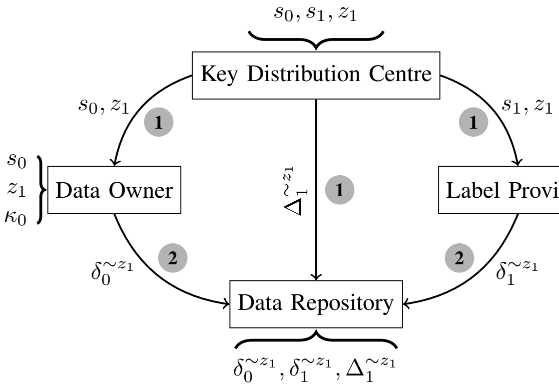


Figure 5. Illustration of the bootstrap sequence of our scheme.

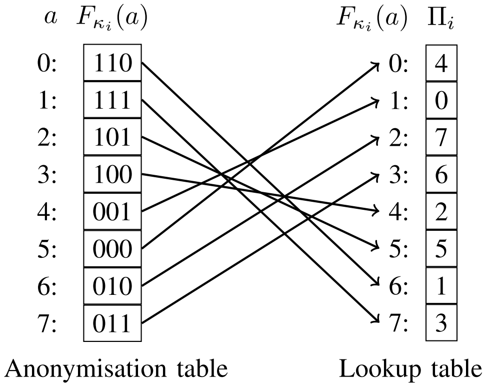


Figure 6. Full lookup table Π_i for permutation $\sim z_i$ of bit length $n = 3$.

the permutation tables sent to the data repository during the bootstrap sequence. More formally, let $a' = F_{\kappa_i}(a)$ denote the anonymised IP address which results from anonymising a using secret κ_i and let $\pi_{i,a'}$ denote the a'-th cell of table Π_i, then we define $\pi_{i,a'} = a^{\sim z_i}$. Informally speaking, $\pi_{i,a'}$ is defined to contain the value resulting from the permutation of a according to z_i. Figure 6 illustrates this process. The values are given in accordance to the example permutation used in Figure 4. As can be seen in this example, assuming $a = 000$ we compute anonymised IP address $a' = F_{\kappa_i}(a) = 110$ which corresponds to decimal value 6. The 6-th cell of table Π_i, i.e. $\pi_{i,6}$, takes decimal value 1. Using the latter as index to the permutation tables illustrated in Figure 4, we note that at this position bit differences of two strings resulting from anonymisation of address $b = 000$ are computed, which exactly corresponds to our assumed address a.

b) Event Correlation: Using lookup table Π_i provided by the i-th label provider together with the information already available after successful bootstrap, the data repository is able to map anonymised IP addresses present in events received from the label provider to IP addresses present in network traces received from the data owner. Consequently, the data repository is able to correlate events and transfer labels associated with events present in E_i. The computation is as follows: let a denote an unknown real IP address for which we want to transfer labels from events in E_i to packets in D and let $a'_{\kappa_0} = F_{\kappa_0}(a)$ denote the corresponding anonymised IP address as received from the data owner and, similarly, $a'_{\kappa_i} = F_{\kappa_i}(a)$ denote the anonymised IP address received from the i-th label provider. From the lookup table we obtain $j = \pi_{i,a'_{\kappa_i}}$, i.e. the correlation index to use in permutation tables $\delta_0^{\sim z_i}, \delta_i^{\sim z_i}$ and $\Delta_i^{\sim z_i}$ and, consequently, cells $\delta_{0,j}^{\sim z_i} = F_{s_0}(a) \oplus F_{\kappa_0}(a)$, $\delta_{i,j}^{\sim z_i} = F_{s_i}(a) \oplus F_{\kappa_i}(a)$ and $\Delta_{i,j}^{\sim z_i} = F_{s_0}(a) \oplus F_{s_i}(a)$. With this, we derive a'_{κ_0} from a'_{κ_i} as

$$a'_{\kappa_0} = a'_{\kappa_i} \oplus \delta_{i,j}^{\sim z_i} \oplus \Delta_{i,j}^{\sim z_i} \oplus \delta_{0,j}^{\sim z_i}. \tag{4}$$

Vice versa, the data repository can derive a'_{κ_i} from a'_{κ_0} as follows:

$$a'_{\kappa_i} = a'_{\kappa_0} \oplus \delta_{0,j}^{\sim z_i} \oplus \Delta_{i,j}^{\sim z_i} \oplus \delta_{i,j}^{\sim z_i}. \tag{5}$$

So far we have demonstrated a process to anonymise IP addresses in a way that allows to correlate two IP addresses which have been anonymised using different secrets. For correlating network events and, especially, for transferring labels, however, we need a specific correlation algorithm which we propose next. Therefore, let $E_i = \{e_{i,1}, e_{i,2}, \ldots, e_{i,m}\}$ denote a set of $m \in \mathbb{N}$ events received by the i-th label provider, with $e_{i,m} = \langle p_{m,1}^{(i)}, p_{m,2}^{(i)}, \ldots, p_{m,q}^{(i)}, l_m^{(i)} \rangle$. Here $p_{m,q}^{(i)}$ denotes the q-th ($q \in \mathbb{N}$) pattern description of the m-th event received from the i-th label provider and $l_m^{(i)}$ denotes the label assigned to event $e_{i,m}$ if all patterns $p_{m,q}^{(i)}$ match. Using this notation a generic event correlation algorithm is given in Figure 7.

This algorithm creates a sample S of traces present in data set D, which is used to extract specific features that are compared against the patterns $p_{m,q}^{(i)}$ associated with event $e_{i,m}$. If all patterns associated with a specific event match the features extracted from S, then S is assigned the label $l_m^{(i)}$ provided by the i-th label provider for event $e_{i,m}$. For brevity, our generic algorithm models the equality of IP addresses present in data set D and event set E_i as a specific pattern. If during pattern processing the algorithm notices that a pattern is an IP address check, the IP address is mapped using our equation (4). For this, permutation and lookup tables as described above are required.

The algorithm we provide here is generic in the sense that it can be applied to single packets or sequences of packets sharing specific attributes (e.g., NetFlow records) as we do not define the specific operation of the SAMPLE procedure. Also, we deliberately choose to not define how to match patterns, i.e. procedure PATTERNMATCH, to the given data set as this may depend on the specific pattern. However, we are convinced that it is easy enough to transfer this generic algorithm into a working program that takes individual considerations into account.

3) Proof: What remains is to formally prove the correctness of our scheme. With correctness, we mean that given permutation tables $\delta_0^{\sim z_i}, \delta_i^{\sim z_i}, \Delta_i^{\sim z_i}$ and lookup table Π_i, a data repository is able to map, i.e. correlate, anonymised IP addresses $a'_{\kappa_0} = F_{\kappa_0}(a)$ and $a'_{\kappa_i} = F_{\kappa_i}(a)$ correctly. In other

```
Require: $\delta_0^{\sim z_i}, \delta_i^{\sim z_i}, \Delta_i^{\sim z_i}, \Pi_i$
procedure MAP($a, i$)
    $j \leftarrow \pi_{i,a}$
    return $a \oplus \delta_{i,j}^{\sim z_i} \oplus \Delta_{i,j}^{\sim z_i} \oplus \delta_{0,j}^{\sim z_i}$
end procedure

procedure CORRELATEEVENTS($D, E_i$)
    $L \leftarrow \emptyset$
    $S \leftarrow$ SAMPLE($D$)
    for $u \leftarrow 1, m$ do
        $c_u \leftarrow 0$
        for $v \leftarrow 1, q$ do
            if ISIPADDRESS($p_{u,v}^{(i)}$) then
                $p_{u,v}^{(i)} \leftarrow$ MAP($p_{u,v}^{(i)}, i$)
            end if
            if PATTERNMATCH($p_{u,v}^{(i)}, S$) then
                $c_u \leftarrow c_u + 1$
            end if
        end for
        if $c_u = q$ then
            $L \leftarrow L \cup \{l_u^{(i)}\}$
        end if
    end for
    ASSIGNLABELS($S, L$)
end procedure
```

Figure 7. Generic event correlation algorithm.

words, we want to prove equation (4), which we do as follows. For the sake of readability we again define $j = \pi_{i,a'_{\kappa_i}}$.

$$\begin{aligned} a'_{\kappa_i} \oplus \delta_{i,j}^{\sim z_i} \oplus \Delta_{i,j}^{\sim z_i} \oplus \delta_{0,j}^{\sim z_i} &= \\ F_{\kappa_i}(a) \oplus F_{s_i}(a) \oplus F_{\kappa_i}(a) \oplus \Delta_{i,j}^{\sim z_i} \oplus \delta_{0,j}^{\sim z_i} &= \\ F_{\kappa_i}(a) \oplus F_{\kappa_i}(a) \oplus F_{s_i}(a) \oplus \Delta_{i,j}^{\sim z_i} \oplus \delta_{0,j}^{\sim z_i} &= \\ F_{s_i}(a) \oplus F_{s_0}(a) \oplus F_{s_i}(a) \oplus \delta_{0,j}^{\sim z_i} &= \\ F_{s_i}(a) \oplus F_{s_i}(a) \oplus F_{s_0}(a) \oplus \delta_{0,j}^{\sim z_i} &= \\ F_{s_0}(a) \oplus F_{s_0}(a) \oplus F_{\kappa_0}(a) &= \\ F_{\kappa_0}(a) &= a'_{\kappa_0} \end{aligned}$$

■

Essentially, our proof is based on properties of the exclusive-or operation, i.e. $g \oplus h = h \oplus g$ (commutative), $g \oplus 0 = g$ (identity element) and $g \oplus g = 0$ (by definition).

IV. Discussion

In this section we discuss design aspects (Sect. IV-A), security properties (Sect. IV-B) as well as practical considerations and implications (Sect. IV-C) of our scheme.

A. Design Aspects

a) Parameters: In contrast to Crypto-PAn [1], our scheme relies on a big amount of information being distributed across all collaborating entities. However, this circumstance is a requirement to not only be able to correlate network events, but also to be able to adhere to the trust model we defined in Section III-B. While we agree that this introduces complexity, we were not able to develop a scheme that would require less parameters while meeting the trust model. On the other hand, we are convinced that adhering to the trust model we set out is a requirement for our scheme to be accepted by data owners. The advantage of being able to correlate events and transfer labels should outweigh the complexity introduced.

From the parameters we exchange, several can be deleted after successful bootstrap. In general, the key distribution centre is not required to store any information at all after the bootstrap sequence has finished. A data owner is not required to store z_i and s_0 after permutation table $\delta_0^{\sim z_i}$ has been transferred to the data repository. Label providers, in contrast, need to securely store parameters z_i and κ_i $(i \geq 1)$ in order to be able to generate lookup table Π_i.

b) Permutation $\sim z_i$: One important parameter is z_i and the permutation it induces. Essentially, one could argue that the security of our scheme depends on the goodness of z_i as $a^{\sim z_i}$, which is stored in lookup table Π_i and transferred to the data repository, may potentially leak information about a. If that would be the case, anonymisation would effectively been broken. On the other hand, as we noted in section II, any prefix-preserving anonymisation function following the canonical form theorem of Xu et al. [1] is essentially a permutation of the IP address space.

We leverage this observation in our scheme by defining the permutation $\sim z_i$ as $F_{z_i}(\cdot)$, i.e. we permute indices of permutation tables $\delta_i^{\sim z_i}, \delta_0^{\sim z_i}$, and $\Delta_i^{\sim z_i}$ using Crypto-PAn and a shared secret z_i. Consequently, $a^{\sim z_i} = F_{z_i}(a)$ and the permutation $\sim z_i$ inherits all security properties of Crypto-PAn, making the permutation as secure as anonymisation currently already applied in practice.

c) Re-keying: When contributing a lot of data to a data repository, a data owner or label provider may be interested in using different keys for anonymisation over time in order to prevent semantic attacks (e.g, [14], [15], [16]) on the anonymisation scheme. Our scheme supports this in two different modes: first, a data owner and every label provider can change its private secret κ_i $(i \geq 0)$ at its sole discretion. What is required in this case, however, is that a new permutation table $\delta_i^{\sim z_i}$ is computed and transmitted to the data repository. Second, re-keying can be coordinated by the key distribution centre if any of s_i, z_i should be renewed. In that case, the whole bootstrap sequence has to be re-iterated.

B. Security Properties

a) Information Leakage: As we set out in earlier sections, the scheme we propose here is completely based on Crypto-PAn [1], the de facto standard in IP address anonymisation: every IP address is anonymised using Crypto-PAn and even the permutation $\sim z_i$, which our scheme relies on, is computed using this algorithm. Hence, our scheme completely inherits security properties of Crypto-PAn with regard to cryptographic attacks, i.e. if a raw/anonymised IP address pair may be available. That is, information about a single address sharing the same $n-1$-bit prefix is leaked in that case, as well as the prefix information.

Nevertheless, due to the nature of collaboration, a label provider is able to derive additional information, and especially to perform semantic attacks, as its labels are transferred to the data set D. This is inevitable and the severity of information leakage depends on the patterns and labels sent. However, the leaked information is only available to the label provider and nobody else. Hence, liabilities may be defined in a collaboration contract between a data owner and label provider that prevent from abuse. Also, one may argue that similar attacks are even possible using plain Crypto-PAn [5].

b) Secret Theft: Like all cryptographic schemes, our scheme is susceptible to secret theft. To challenge attackers, we designed our scheme such that no compromise of any single identity will give an attacker the ability to de-anonymise data present in the repository. However, if an attacker is able to compromise any two entities and to steal secrets, then our scheme can effectively be de-anonymised.

C. Practical Considerations

a) Freedom of Collaboration: Especially from a data owner's perspective, an anonymisation scheme should be designed such that the data owner can decide who to grant access to raw IP address information, i.e. not anonymised IP addresses, and with whom a data owner collaborates. Especially the latter is important for our scheme, as data owners may not universally trust any label provider. This freedom of collaboration is embedded in our scheme as a data owner must actively participate in the bootstrap sequence in order to distribute all required information. If a data owner does not want to collaborate with a specific label provider, he simply does not compute permutation table $\delta_0^{\sim z_i}$. Similarly, if a data owner would like to stop collaboration with a label provider, the data owner re-keys κ_0 and does not send an updated $\delta_0^{\sim z_i}$ to the data repository.

b) Co-locating Key Distribution Centre with Data Repository: Co-locating the key distribution centre with the data repository may be an option in practice. In that case, however, it is of utmost importance that secrets s_0, s_i $(i \geq 1)$ are stored securely and independent of the data sets D, E_i and Π_i. Otherwise, compromise of the data repository reveals enough information to de-anonymise IP addresses present in the data set. In case such co-location is envisaged, we recommend to choose secrets s_0, s_i, z_i and to compute $\Delta_i^{\sim z_i}$ offline using a specially secured host and to immediately erase parameters s_0, s_i, z_i after computation of $\Delta_i^{\sim z_i}$.

c) Efficiency: Our scheme is rather time efficient. Using commodity hardware, we were able to compute permutation tables for the full IPv4 address space (i.e., IP addresses of length $n = 32$ bit) in less than 5 minutes. Regarding space efficiency, for IPv4 our scheme requires 16 GiB (2^{32} addresses $\times$ 32 bit / address) at the data repository per collaborator in order to store permutation tables. That is, in total 48 GiB have to be stored on the data repository in order to be able to anonymise and correlate events that span across full IPv4 address space per collaboration, i.e. 16 GiB for each permutation table. For a data repository, this amount of space is negligible as network traces captured on a contemporary 10 Gbit/s ethernet port with 50% utilisation result in an equivalent amount of data in less than 90 seconds. We argue that sacrificing 90 seconds of network traces in favour of our scheme is tolerable. Additionally, for the case that it is known in advance that only a fraction of IP addresses will be present in D, our scheme can be adapted to compute sparse tables only, which linearly reduces the required disk space.

V. Related Work

Sharing and anonymising network traces are old, but steady and, essentially, very important research topics.

In [17], Allman and Paxson discuss ethics and general issues of sharing measurement data and provide considerations for data providers as well as data receivers. Along this line, Porras and Shmatikov [4] discuss risks and challenges associated with large-scale data collection and sanitisation. Allman et al. [18] describe a scalable system for sharing Internet measurement data. With PS2, a privacy-sensitive sharing framework is proposed by Kenneally and Claffy [19]. In earlier work [10], we propose and motivate a data sharing codex for research.

In addition to Crypto-PAn [1], Minshall [20] provides the utility TCPdpriv, which can be used to remove private information from packet captures in pcap file format. Lincoln et al. [21] propose to hash IP addresses using HMAC [22] and SHA-1 [23] hash functions, depending on context. Anonymisation using bloom filters [24] is proposed by Locasto et al. [25]. In [3], Xu and Ning propose to anonymise data using concept hierarchies, which can be used for probabilistic correlation. Pang et al. [26] provide a high-level programming language for network trace anonymisation as extension to Bro IDS [27].

VI. Conclusion and Future Work

In this work we proposed an IP address anonymisation scheme based on Crypto-PAn, the de facto standard in IP address anonymisation, that can be used to anonymise IP addresses such that events observed in one data set can be correlated with network packets present in another data set without sharing the same anonymisation keys. Our scheme is novel and not only effectively facilitates sharing labels in order to improve the availability of reference data in network research, but also facilitates network management by allowing to, for instance, share and correlate incidents without revealing IP addresses. The scheme we propose is secure in its best possible way: it inherits, without compromise, all security properties of Crypto-PAn, but is susceptible to rogue collaborators and large-scale theft of cryptographic secrets. Furthermore, our scheme works efficiently for IPv4 addresses, but, unfortunately, not for IPv6. For the latter our scheme can not efficiently be applied as delta and permutation tables can not efficiently be computed and stored. The study on how our scheme can be adapted such that it scales to IPv6 is left for future work.

References

[1] J. Xu, J. Fan, M. H. Ammar, and S. B. Moon, "Prefix-preserving ip address anonymization: Measurement-based security evaluation and a new cryptography-based scheme," in *Proceedings of the 10th IEEE International Conference on Network Protocols*. IEEE, 2002, pp. 280–289.

[2] D. Koukis, S. Antonatos, and K. G. Anagnostakis, "On the privacy risks of publishing anonymized ip network traces," in *Proceedings of the 10th IFIP TC-6 TC-11 International Conference on Communications and Multimedia Security*, ser. CMS'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 22–32.

[3] D. Xu and P. Ning, "Privacy-preserving alert correlation: a concept hierarchy based approach," in *Proceedings of the 21st Annual Computer Security Applications Conference*. IEEE, 2005.

[4] P. Porras and V. Shmatikov, "Large-scale collection and sanitization of network security data: Risks and challenges," in *Proceedings of the 2006 Workshop on New Security Paradigms*, ser. NSPW '06. New York, NY, USA: ACM, 2007, pp. 57–64.

[5] S. E. Coull, C. V. Wright, F. Monrose, M. P. Collins, M. K. Reiter *et al.*, "Playing devil's advocate: Inferring sensitive information from anonymized network traces." in *Proceedings of the 2007 Annual Network and Distributed System Security Symposium*, vol. 7. ISOC, 2007, pp. 35–47.

[6] S. Sheng, B. Wardman, G. Warner, L. F. Cranor, J. Hong, and C. Zhang, "An empirical analysis of phishing blacklists," in *Proceedings of the 6th Conference on Email and Anti-Spam*, 2009.

[7] A. Ramachandran, D. Dagon, and N. Feamster, "Can dns-based blacklists keep up with bots?" in *Proceedings of the 3rd Conference on Email and Anti-Spam*, 2006.

[8] C. Kreibich and J. Crowcroft, "Honeycomb: creating intrusion detection signatures using honeypots," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 1, pp. 51–56, 2004.

[9] G. Portokalidis, A. Slowinska, and H. Bos, "Argos: an emulator for fingerprinting zero-day attacks for advertised honeypots with automatic signature generation," *ACM SIGOPS Operating Systems Review*, vol. 40, no. 4, pp. 15–27, 2006.

[10] S. Abt and H. Baier, "A darknet-driven approach to compilation of hostile network traffic samples," in *Proceedings of 20th DFN Workshop zu Sicherheit in vernetzten Systemen*. BoD – Books on Demand, 2013, pp. E1–E21.

[11] C. Fachkha, E. Bou-Harb, A. Boukhtouta, S. Dinh, F. Iqbal, and M. Debbabi, "Investigating the dark cyberspace: Profiling, threat-based analysis and correlation," in *Proceedings of the 7th International Conference on Risk and Security of Internet and Systems*. IEEE, 2012, pp. 1–8.

[12] S. Abt and H. Baier, "Are we missing labels? A study of the availability of ground-truth in network security research," in *Proceedings of 3rd Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*. IEEE, 2014.

[13] J. Daemen and V. Rijmen, *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media, 2013.

[14] M. Burkhart, D. Schatzmann, B. Trammell, E. Boschi, and B. Plattner, "The role of network trace anonymization under attack," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 5–11, 2010.

[15] T. Brekne, A. , and A. ø, "Anonymization of ip traffic monitoring data: Attacks on two prefix-preserving anonymization schemes and some proposed remedies," in *Proceedings of the 5th International Conference on Privacy Enhancing Technologies*, ser. PET'05. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 179–196.

[16] T. Brekne and A. Årnes, "Circumventing ip-address pseudonymization," in *Proceedings of the Third IASTED International Conference on Communications and Computer Networks, October 24-26, 2005, Marina del Rey, CA, USA*. IASTED/ACTA Press, 2005, pp. 43–48.

[17] M. Allman and V. Paxson, "Issues and etiquette concerning use of shared measurement data," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. ACM, 2007, pp. 135–140.

[18] M. Allman, E. Blanton, and W. Eddy, "A scalable system for sharing internet measurements," in *Proceedings of the 2002 Workshop on Passive and Active Measurements*, 2002.

[19] E. Kenneally and K. Claffy, "Dialing privacy and utility: A proposed data-sharing framework to advance internet research," *IEEE Security Privacy*, vol. 8, no. 4, pp. 31–39, July 2010.

[20] G. Minshall, "Tcpdpriv," 1997. [Online]. Available: http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html

[21] P. Lincoln, P. A. Porras, and V. Shmatikov, "Privacy-preserving sharing and correlation of security alerts." in *Proceedings of the 2004 USENIX Security Symposium*. USENIX, 2004, pp. 239–254.

[22] H. Krawczyk, R. Canetti, and M. Bellare, "Hmac: Keyed-hashing for message authentication," 1997.

[23] D. Eastlake and P. Jones, "Us secure hash algorithm 1 (sha1)," 2001.

[24] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, Jul. 1970.

[25] M. E. Locasto, J. J. Parekh, A. D. Keromytis, and S. J. Stolfo, "Towards collaborative security and p2p intrusion detection," in *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*. IEEE, 2005, pp. 333–339.

[26] R. Pang and V. Paxson, "A high-level programming environment for packet trace anonymization and transformation," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2003, pp. 339–351.

[27] V. Paxson, "Bro: a system for detecting network intruders in real-time," *Computer networks*, vol. 31, no. 23, pp. 2435–2463, 1999.