Adversarial Matching of Dark Net Market Vendor Accounts

Xiao Hui Tai
Carnegie Mellon University
xtai@cmu.edu

Kyle Soska
Carnegie Mellon University
ksoska@cmu.edu

Nicolas Christin
Carnegie Mellon University
nicolasc@cmu.edu

ABSTRACT

Many datasets feature seemingly disparate entries that actually refer to the same entity. Reconciling these entries, or "matching," is challenging, especially in situations where there are errors in the data. In certain contexts, the situation is even more complicated: an active adversary may have a vested interest in having the matching process fail. By leveraging eight years of data, we investigate one such adversarial context: matching different online anonymous marketplace vendor handles to unique sellers. Using a combination of random forest classifiers and hierarchical clustering on a set of features that would be hard for an adversary to forge or mimic, we manage to obtain reasonable performance (over 75% precision and recall on labels generated using heuristics), despite generally lacking any ground truth for training. Our algorithm performs particularly well for the top 30% of accounts by sales volume, and hints that 22,163 accounts with at least one confirmed sale map to 15,652 distinct sellers—of which 12,155 operate only one account, and the remainder between 2 and 11 different accounts. Case study analysis further confirms that our algorithm manages to identify non-trivial matches, as well as impersonation attempts.

KEYWORDS

Record linkage; Dark net; Adversarial classification; Measurements

ACM Reference Format:
Xiao Hui Tai, Kyle Soska, and Nicolas Christin. 2019. Adversarial Matching of Dark Net Market Vendor Accounts. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3292500.3330763

1 INTRODUCTION

Many datasets feature seemingly disparate entries that actually refer to the same real-world entity. For instance, in a demographic census, a unique individual may appear under several entries ("John Public," "John Q. Public," etc.) that have to be reconciled prior to analysis to ensure the quality of the census data. This reconciliation process is typically termed *matching*, and has been extensively studied in statistics, commonly under the name *record linkage* [6]. Other matching instances include reconciling disparate casualty reports during disasters or wars, linking census records to other demographic surveys, disambiguating inventors, authors, or movies in patent, bibliographic, or movie databases.

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08.
https://doi.org/10.1145/3292500.3330763

Matching is challenging due to error—including approximations during data collection. Fields between different datasets may be different, and entries may be missing fields. Many matching algorithms tackle these problems, but most do not consider that the presence of separate entries in the original data may be due to malice. In some contexts, however, *adversaries* have a vested interest in seeing the matching process fail, either by finding spurious matches, or by failing to match entries that refer to the same entity.

An example of such an *adversarial matching* context is the online anonymous marketplace ("dark net market") ecosystem [5, 7, 11, 14, 18, 22, 25, 30, 34]. Different from traditional electronic commerce marketplaces such as Amazon Marketplace, eBay, or AliBaba, online anonymous marketplaces strive to offer *anonymity* guarantees to both buyers and sellers, by relying on network anonymizers (e.g., Tor [10]), and, increasingly, on privacy-minded cryptocurrencies such as Monero [21]. Regrettably, a significant fraction of online anonymous marketplace transactions involve illicit goods [25].

In such an anonymous setting, vendors who operate different accounts may not want these different accounts to be matched. For instance, a given vendor may compartmentalize different lines of businesses between different accounts to increase their operational security [27]. Conversely, some vendors may attempt to impersonate well-known accounts—e.g., by registering the same handle on a different marketplace—to defraud unsuspecting customers.

Coming up with proper matching techniques is important for researchers, law enforcement, and patrons of these marketplaces. Researchers studying these ecosystems might be interested in getting an accurate idea of the number of sellers involved. Any analysis related to seller behavior is more suitably done on a seller-level than an account-level, for example seller longevity and reputation. Law enforcement have an interest in linking online identities to real-world identities, and information from different accounts can be combined to provide leads.[1] In the absence of an automated matching technique, there have been reports of law enforcement using online forums or manual comparisons of items sold [29], as well as the Grams search engine [3, 28].

Our problem shares some similarities with the general issue of *Sybil* detection in distributed systems [12]. Sybils are accounts that are all controlled by the same person or group to gain an advantage; for instance, to manipulate reputation in online reviewing systems or promote social networking accounts [32]. The key difference, though, is that Sybil operators generally need to control many accounts to achieve their objectives and as a result, Sybil detection can benefit from observing common patterns among Sybils. By contrast, we deal with fewer accounts which are truly operated by humans, and thus emit less regularity in their observable patterns.

[1] For example, in perhaps the most well-known marketplace-related arrest, the creator of Silk Road was known by a pseudonym, Dread Pirate Roberts. Authorities linked this to another account on a message board, Frosty, and Frosty's real-world identity was known to be Ross Ulbricht, contributing to his eventual arrest [23].

Furthermore, Sybil detection algorithms usually do not try to address the dual issue of impersonation attacks. In this respect, our work is more closely related to detection of "sockpuppet" accounts (a user account controlled by an individual who controls at least one other user account) in online discussion forums [17].

There have been a few previous attempts to match vendor accounts in online anonymous marketplaces. Two took the simple approach of matching (PGP) cryptographic public keys advertised by different accounts [5, 25] which cannot handle impersonation attacks, since anybody can copy somebody else's public key. At the other end of the complexity spectrum, the Grams search engine [3] was an elaborate and largely manual and crowdsourced attempt to match accounts across marketplaces. Grams was taken offline in December 2017, reportedly due to the high human cost of operating such a database [2]. Furthermore, crowdsourcing is vulnerable to poisoning, in which an adversary injects malicious content.

In this paper, we describe an automated classifier that matches disparate vendor accounts and detects impersonators, relying on features that are cumbersome for an adversary to forge. We evaluate on eight years (2011–2018) of online anonymous marketplace data, and find that 22,163 accounts with at least one confirmed sale map to 15,652 distinct sellers. 12,155 sellers (77%) operate only one account, while the remainder operates between 2 (1,909 sellers) and 11 accounts (2 sellers). Because ground truth in this context is elusive, we compare the performance of our algorithm with data obtained from Grams and from PGP key matching—although these labels are problematic, they give a general sense of model performance. Using these labels, our algorithm achieves more than 75% precision and recall. It works particularly well for vendor accounts with significant sales volume (approximately 90% recall at 75% precision for the top 30% of accounts). We present a few case studies where ground truth is documented through criminal complaints or forum discussions; we can automatically discover reported impersonation attempts and non-trivial links between accounts. We also discuss scalability limitations of our matching algorithm. Our hope is to pave the way for additional research in the context of adversarial matching, while giving the research community access to our data.

2 BACKGROUND AND RELATED WORK

We focus our literature review on three complementary areas: statistics research on matching problems, Sybil detection in social networks and distributed systems, and online anonymous marketplace measurements.

Matching. Matching in computer science and statistics refers to the disambiguation of records in the absence of a unique or common identifier. Probabilistic methods estimate, for each pair of records, a match probability based on available features. If the probability of match exceeds some cutoff, the pair is identified to be a match. Fellegi and Sunter's 1969 probabilistic framework [13] is often considered the standard model for unsupervised record linkage [35]. When labels are available, the matching problem can also be seen as a supervised classification task, and more recently, machine learning methods have gained popularity (see, e.g., [31]). In the literature, records often refer to single entries in demographic, bibliographic, and other databases [6]. By contrast, we treat records as an entire account profile, consisting of a history of user pages, inventories and sales. Further, our application to dark net marketplaces has a unique adversarial component, that to our knowledge has not been studied in detail in the matching literature.

Sybil and impersonator detection. Douceur [12] formalized the Sybil attack in the context of peer-to-peer networks. Sybil attacks frequently operate at scale, which requires a degree of automation. Subsequent work proposed defenses against Sybil attacks in social networks [33] and online reviewing systems [32] by exploiting regularity in certain patterns (e.g., social network account creation date, number and type of followers) that automated Sybil creation produce in order to distinguish between Sybils and human-operated accounts. Our matching problem is different: the accounts we need to match are all curated and operated by humans. We also have to address impersonation attacks, in which an individual creates an account with the fraudulent goal of impersonating a different vendor. Impersonation attacks have, for instance, been studied in the context of DNS typosquatting [1, 16, 20, 26]. Carried out at scale, these attacks can be detected by analyzing relatively simple features, whereas we have to consider more targeted, smaller-scale attempts. As mentioned in the introduction, our work more closely resembles Kumar et al.'s [17] work on "sockpuppets" in online discussion communities. The context they analyze, however, calls for markedly different features, e.g., writing style and community interactions such as responses to posts. Their focus is also on characterizing behavior as opposed to the actual matching task.

Online anonymous marketplaces. Researchers have tried to model the economics of online anonymous marketplaces ("dark net markets"). This began with descriptive statistics of the Silk Road marketplace [7], later expanding to include characterizations of the whole ecosystem and its longitudinal evolution [25], as well as analyses of specific categories of goods [14, 30]. These papers mostly focus on macro-level considerations—e.g., calculating the total revenue of a given marketplace or of a given line of business over time—which do not require disambiguation between vendor accounts. A smaller number of papers have attempted to characterize online anonymous marketplaces at a finer, transaction-level granularity, for instance to evidence geographic properties of the trade [11, 22]. These studies do not attempt to link vendor accounts.

Closer to our own efforts, Broséus et al. [5] analyzes vendor activity across eight different marketplaces. Like Soska and Christin [25], Broséus et al. rely on PGP keys for account matching, which, as we discussed in the introduction, is an imperfect proxy. Wang et al. [34] attempt to perform account linkage by analyzing item images. While a promising direction, image features could be changed at relatively low cost for an attacker, or normalized by a marketplace operator prior to publication. Finally, recent work in cryptocurrency tracing tries to link different financial accounts owned by the same entities by examining the underlying cryptocurrency ledgers [15, 19, 21]. We complement these approaches by matching vendor accounts solely using publicly available marketplace data.

3 DARK NET MARKET DATASET

We combine data collected by Soska and Christin [25] (including data from the original 2013 Silk Road study [7]), with data collected for the AlphaBay marketplace [21, 30], and finally with data more

Market	# snap.	Collection interval	# accts w/ sale	# accts w/ key	# keys
Silk Road 1	164	2011-22-11 to 2013-08-18	2,327	467	574
BMR	25	2013-10-11 to 2013-11-29	975	0	0
Silk Road 2	195	2013-11-24 to 2014-10/26	1,196	1,359	1,926
Pandora	140	2013-12-01 to 2014-10-28	457	0	0
Agora	161	2013-12-28 to 2015-06-12	1,956	2,563	3,246
Hydra	29	2014-07-01 to 2014-10-28	132	10	11
Evolution	45	2014-07-02 to 2015-02-16	2,338	11,586	12,288
AlphaBay	27	2015-03-18 to 2017-05-24	6,215	8,370	9,865
Dream	19	2017-07-15 to 2018-08-20	4,305	3,950	4,281
Valhalla	3	2017-07-28 to 2017-12-06	268	332	341
Traderoute	5	2017-07-28 to 2017-10-11	1,768	2,463	2,592
Berlusconi	8	2017-11-22 to 2018-08-22	226	0	0
Other	175	2013-10-19 to 2014-08-11	N/A	999	1,160
Total	*996*	*2011-22-11 to 2018-08-22*	*22,163*	*32,101*	*36,284*

Table 1: Markets collected and analyzed. Markets are chronologically ordered.

recently collected from the Dream, Berlusconi, Valhalla, and Traderoute marketplaces. We obtained these data by repeated scraping and parsing of all web pages present in these marketplaces. We refer the reader to our previous work [7, 25], for a discussion of the technical and ethical details of data collection.

Most marketplaces consist of a collection of publicly-accessible vendor profile and item pages. Vendor pages contain a description of the vendor, including their various offerings, and, quite frequently, a cryptographic (PGP) public key which can be used by buyers to encrypt and authenticate communications with the vendor.

Item pages describe a given product, its associated price, and feedback left by buyers about their purchases of the item. As prior work [25] has shown, these items can generally be characterized into just a few high level categories. Feedback is often, but not always, mandatory, and is a reasonable proxy for sales [7, 25].

3.1 Data corpus overview

Table 1 describes, for each marketplace we consider, the number of snapshots taken, the data collection interval, the number of vendor accounts, and the number of distinct PGP keys present on the marketplace. PGP keys extraction is imperfect, as evidenced by the absence of keys extracted from Berlusconi, BMR, and Pandora. "Other" corresponds to smaller markets (Flo, Utopia, The Marketplace, Tor Bazaar) or markets for which our data is very sparse (Sheep). These are only used to cull additional PGP keys. Evolution data also contains several non-vendor accounts.

For analyzing PGP keys and labeling data we consider the entire dataset of 178,270 accounts. However for training and evaluating our model (Sections 5 and 6) we only consider vendor accounts with at least one publicly reported sale (i.e., at least one piece of feedback). This for instance means that we discard 3,613 zero-sale accounts from AlphaBay.

3.2 Account-level information

For each vendor account, we extract information, such as their ID, sales category and diversity, profile data, item title and descriptions, and feedback received. We include characteristics that are difficult for an adversary to control, such as item prices and days in which sales were made.

We also extract per-account inventories of items that had at least one sale throughout the period. For each item, we extract the predicted category [25], the dosage (number and unit, e.g. "8 grams"), and the quantity e.g., number of pills, tabs, tablets, blotters, etc. To infer dosages and quantities, we use regular expressions [8].

Finally, we extracted 28,417 PGP keys from the profiles and item listings of 32,101 distinct accounts. Because some vendors use the same key on multiple accounts on different marketplaces, there are significantly more accounts with a PGP key than unique PGP keys. For most modern marketplaces, the number of accounts with a PGP key is similar to those with a sale. Evolution is an exception: the data contains non-vendor (buyers or simply curious individuals) accounts that may contain PGP keys.

3.3 Grams data

Grams, the "dark web search engine [3]" was an attempt at linking and supplementing different vendor profiles on various marketplaces. The procedure by which vendor profiles were linked was never clear, but clues point to extensive manual curation. When Grams shut down in December 2017, its purported administrators released their vendor databases to the public [2]. We use these databases as an additional source of information.

Grams reports knowledge of about 38,416 handles mapping to 27,491 unique vendors, over 15 marketplaces, and associated with 22,357 unique PGP keys. Five of the marketplaces Grams includes overlap with ours: Agora, AlphaBay, Evolution, Valhalla, Dream. The data correspond to 28,727 handles, grouped into 19,021 unique vendors which collectively published over 19,957 PGP keys. The Grams database contains an `impostor` field to denote impersonation attacks. This field is set on 129 handles, including 102 handles for the marketplaces of interest to us.

3.4 Discussion

As can be expected from such a longitudinal collection effort, our data is incomplete. For instance, no PGP public key was extracted from our Berlusconi snapshots, despite evidence that as many as 52 different keys were present on the site at the time(s) we scraped it. However, for our purposes, completeness of the data corpus is less important than correctness. In fact, to be practical, our matching algorithms should work with incomplete datasets.

4 DATA LABELING

By definition, we do not generally have any ground truth information on matches between handles.[2] Instead, we create three sets of labels from heuristics, based on our existing data. We then discuss properties of the labeled sets.

4.1 Labeling heuristics

The first two labeled sets are derived from PGP key information and the third from Grams data. None of these techniques are perfect, but these labeled sets are be useful for training and comparison with our own classifier, which we describe in Section 5.

[2] In a handful of cases, as we will see in Section 6, ground truth is available from arrest records—but those are only available in a very small fraction of all accounts, and may not be complete even for these accounts.

4.1.1 Common PGP keys. A common technique is to link handles that purport to use the same PGP key [5, 25]. As mentioned briefly in Section 1, this technique does not preclude incorrect links. First, an impersonator can simply cut and paste a public key onto their vendor page. While decrypting or signing a message requires the associated private key, users cannot perform this verification non-interactively. Likewise, two handles using different public keys could belong to the same vendor: people frequently lose access to their private PGP key, and thus have legitimate reasons to generate new PGP keypairs. With these caveats in mind, for this first set of labels, we use the following definition.

Definition 4.1. For any two vendor handles i and j, with associated sets of public PGP keys K_i and K_j, we consider i and j as mapping to the same vendor if and only if $\exists k \in K_i, \exists k' \in K_j$ such that $k = k'$.

The relation defined in 4.1 fails to capture instances where a vendor may have three or more accounts that do not all use the same PGP key but may be linked through an intermediate key. To address this case, we also consider the transitive closure of the relation defined by 4.1.

4.1.2 Signature rings. PGP keys are associated with "user ids," which typically take the form of an email address. PGP offers the ability for a user to *sign* other users' public keys with their private key. This signature operation is an endorsement that the signed key is valid, and matches the user id it purports to belong to. In other words, if user A signs user B's key, A asserts that B's key does indeed belong to B. We thus use the following definition for our second set of labels.

Definition 4.2. For any two vendor handles i and j, with associated sets of public PGP keys K_i and K_j, we consider i and j map to the same vendor if and only if $\exists k \in K_i, \exists k' \in K_j$ such that 1) the private key R_k associated with k signs k', and 2) the email addresses associated to k and k', $e(k)$ and $e(k')$, match (i.e., $e(k) = e(k')$).

Definition 4.2 should theoretically encompass Definition 4.1 as users are expected to always sign their own key; we were surprised to discover that most users in this corpus actually do not do this.

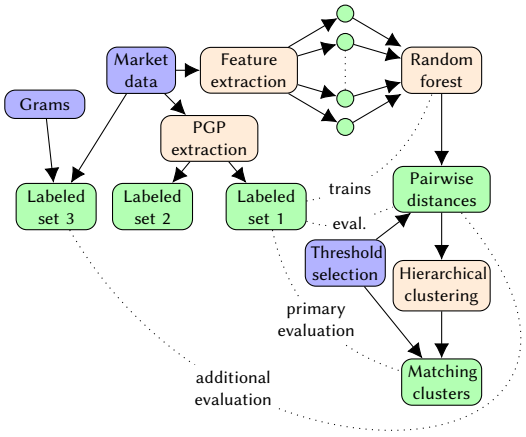
4.1.3 Grams data. The third set of labels we consider comes from Grams. For each handle i, the Grams database contains a `link` identifier. Different handles sharing the same link reportedly belong to the same vendor. Formally,

Definition 4.3. For any two vendor handles i and j, we consider i and j map to the same vendor if and only if $\text{link}(i) = \text{link}(j)$.

Definition 4.3 *should* also encompass Definition 4.1 as PGP matching seems to have been one of the criteria Grams used to match users. In practice, though, the sources of data do not necessarily overlap—we have data from marketplaces that Grams does not cover and vice-versa—so we expect certain differences. While Grams intuitively should provide a superior way of labeling handles, compared to PGP linkage, the data Grams provide are limited to pre-2018 records only, miss a number of marketplaces we are considering, and also struggle with small vendors that only have a few sales.

We also experimented with other possible labeling techniques, such as inferring matches from vendor self-descriptions—but this is infeasible using simple heuristics, and more complex analysis is subsummed by the features we eventually considered.

Figure 1: Classifier overview. This diagram shows the relationship between our classifier, and the datasets described in Section 4. Blue boxes denote input data; green boxes, outputs; and salmon boxes, computations. Arrows denote input-output relations.



4.2 Labeled set properties

The relation of Definition 4.1 finds 13,099 pairwise relations on 11,711 out of the 32,101 vendor accounts that posted a key in Table 1. The transitive closure of this relation yields another 855 pairs.

Despite 3,168 vendors using multiple PGP keys on the same profile, and several more vendors using multiple keys across different profiles, we only observed three instances of the forward signing described in Definition 4.2. One instance was from a PGP key posted on two different accounts on The Marketplace, where one was clearly marked as a test account. Another was from a vendor by the same handle on the Dream, AlphaBay and Traderoute markets, and the last was from keys sourced from the same handle on AlphaBay. So, while in principle, PGP key signing is a powerful technique for asserting the authenticity of a user-key relationship, in practice, vendors do not seem to adopt it at all. This further motivates linking techniques which do not rely on keys.

The addition of the Grams data (Definition 4.3) yielded an additional 18,215 pairwise relations beyond Definition 4.1. This is not surprising since the Grams data is manually curated and identifies instances where vendors either do not use PGP, or have elected to create new keys for their different profiles.

5 CLASSIFIER DESIGN

Figure 1 presents an overview of our system. As mentioned, we first restrict our data to accounts that reported *at least* one sale, that is, accounts that have obtained at least one piece of feedback on one of the item listings they offer. Out of the original 178,270 accounts we originally identified, this corresponds to 22,163 accounts (see Section 3). The labeled sets on the left side of the diagram represent the labeling operations discussed in Section 4. Labeled set 1 corresponds to Definition 4.1, Labeled set 2 corresponds to Definition 4.2, and Labeled set 3 is the combination (union) of Definition 4.1, and Definition 4.3, including transitive closures. The right side of the

diagram describes our classifier system. We extract a number of features from vendor account data and pass them to a random forest classifier. The random forest classifier outputs pairwise distances between accounts. Our classifier might produce matches that are intransitive, which we resolve with hierarchical clustering. We next turn to the discussion of our feature set, and our clustering choices.

5.1 Feature extraction

Our set of classifier features must fulfill three objectives. They must be derivable from publicly-available data; they must (as a set) properly discriminate between matches and non-matches; and they must (as a set) be resilient to evasion or poisoning. That is, an adversary should not be able to easily produce misleading feature sets or copy feature sets from a different vendor. For instance, an account handle is easy to impersonate. If a famous vendor has not yet opened an account on a brand new marketplace, an impersonator could easily create an account with the famous vendor's handle. On the other hand, it is difficult for an adversary to convincingly mimic the sales volume of an established vendor.

We generate pairwise comparisons for the 22,163 accounts, resulting in approximately 245 million pairs. For each pair, from account-level information (see Section 3.2), we compute the following similarity measures: Edit distance between the IDs; Same or different marketplace; Jaccard similarity ($J(A,B) = |A \cap B|/|A \cup B|$ for sets A and B) between bag-of-words representation of profile descriptions, item titles and descriptions (excluding any extracted PGP keys); Inventory-related Jaccard similarities: consider unique categories of all items sold, (category, dosage) pairs, (category, unit) pairs, and (category, dosage, unit) tuples; Absolute difference between diversity coefficients; Absolute difference between number of tokens in the bag-of-words representations of profiles and item descriptions; Absolute difference between number of days active (defined as the period between which sales are recorded); Absolute difference between number of listings with feedback, number of feedback, and number of feedback normalized by days active and marketplace total; Absolute difference between five-number summary of sales prices; Hamming distance ($d_H(x,y) = \frac{1}{n}\sum_{i=1}^{n} I(x_i \neq y_i)$, where x and y are length n vectors) between binary vectors encoding days in which sales occur; Fraction of overlapping sales days (number of days where both accounts have sales / size of union of sales days); Sum of number of sales days for both accounts.

The measures we selected should, as a whole, be costly for an adversary to forge. Because we only consider items which received feedback, an impersonator would not only have to post similar items, but receive sales on these items, or have the ability to generate fraudulent feedback to pretend sales occurred. While the latter is not difficult to do, it is also usually noticed quite quickly by marketplace operators and customers, and offenders are rapidly banned. Most of the other features are directly related to sales as well.

5.2 Training and classification

We use a subset[3] of Labeled set 1 in Figure 1, following Definition 4.1, to train our classifier. This labeled set contains 3,653 matches (ones: the pair of accounts shares at least one PGP key), 73,449,607 non-matches (zeros; the pair of accounts does not share any PGP key), and 172,134,943 pairs have missing labels (at least one of the two accounts considered is not associated with any PGP key). Crucially, we do *not* take these labels as ground truth, since, as discussed in Section 4, they could be vulnerable to impersonation attacks. Instead, for the training set, as an additional step we split this into 10 folds and obtain predictions by training on 9 folds and predicting on the tenth (in the same way that cross-validation is typically done). In this manner, if a training example was a pair consisting of an impersonator copying another vendor's PGP key and that said vendor, the training label would be "match," but it could still have a low model prediction.

We then treat our classification task as a supervised learning problem. We use a random forest classifier based on the extracted features and the labels, and produce a set of pairwise distances. More precisely, the output of the random forest classifier, for any pair of accounts (i, j) is a proportion of votes p_{ij} for the accounts correspond to the same vendor, where p_{ij} = Number of trees voting for match label / Total number of trees in random forest. We compute the "distance" (or dissimilarity) d_{ij} between both accounts as $d_{ij} = 1 - p_{ij}$.

5.3 Hierarchical clustering

Because pairs are evaluated independently, our classifier might produce matches that are intransitive. To resolve this, we use hierarchical agglomerative clustering. Given accounts $1, \ldots, n$, dissimilarities d_{ij} between each pair i and j, and dissimilarities $d(G,H)$ between groups of accounts $G = \{i_1, i_2, \ldots i_r\}$ and $H = \{i_1, i_2, \ldots, i_s\}$, the algorithm starts with each node in a single group, and repeatedly merges groups such that $d(G,H)$ is within a threshold D.

We consider four different linkage methods to define $d(G,H)$. *Single linkage* specifies that $d_{\text{single}}(G,H) = \min_{i\in G, j\in H} d_{ij}$. Informally, each account needs to be matched with only one other account in the cluster, and missing links are filled in. *Complete linkage* uses $d_{\text{complete}}(G,H) = \max_{i\in G, j\in H} d_{ij}$. That is, every account in the cluster needs to match with all other accounts in the cluster, and links are cut if this property is not satisfied. *Average linkage* [24] relies on $d_{\text{average}}(G,H) = \frac{1}{|G||H|}\sum_{i\in G, j\in H} d_{ij}$, which has no intuitive interpretation. Finally, *minimax linkage* [4] uses $d_{\text{minimax}}(G,H) = \min_{i\in G\cup H} r(i, G\cup H)$, where $r(i,G) = \max_{j\in G} d_{ij}$, the radius of a group of nodes G around i. Informally, each account i belongs to a cluster whose center c satisfies $d_{ic} \leq D$; that is, all accounts in the cluster need to match to the cluster center.

6 PERFORMANCE EVALUATION

First, we evaluate performance with respect to various modeling choices. Even though the labels we have defined in Section 4 do not represent ground truth, we believe they are adequate enough to illuminate the effects of various parameter choices. We then apply our classifier to a series of case studies, to see how it performs in practice.

[3] Two authors of this paper extracted PGP keys independently, producing near-identical results. For training, we ended up using a subset of the keys from data earlier in the collection period (these contained 3,653 PGP matches, while the full Labeled set 1 has 7,564 matches). Even with this restricted set of labels, when we evaluate using the full set of PGP keys and Grams labels, including transitive closures, results are reasonable, as we see in Figure 3.

6.1 Classifier accuracy

We first evaluate the random forest classifier. This is done at a pairwise level, predicting whether each pair belongs to the same seller. Because of the large class imbalance (most account pairs do not match), we avoid measures that use the number of true negatives. Instead of using traditional receiver operating characteristic (ROC) curves, we graph precision (True Positive/Predicted Positive) versus recall (True Positive/Actual Positive).

Using the training data (described in Section 5), we train the model sampling all of the labeled matches, and between 3,653 to 10 million labeled non-matches. We then predict on all remaining non-sampled pairs. We additionally split the sampled pairs into 10 folds and obtain predictions on these by training on 9 folds and predicting on the tenth. We evaluate performance first with respect to the labeled set. As detailed in Section 5, this consists of 73,453,260 pairs (3,653 labeled matches), coming from the 12,121 accounts that posted at least one PGP key. The remaining pairs are discarded in this evaluation. In addition, we compared model performance against a baseline of classifying pairs solely using a threshold-based approach on edit distances of the IDs.

Figure 2 summarizes the results for three types of experiments. Figure 2(a) shows that a classifier sampling at least 30,000 non-matches strictly outperforms the baseline of ID distance. Also, trying to increase recall past 80-85% decreases precision dramatically. In other words, trying to correctly predict all actual matches results in true non-matches overwhelmingly being predicted to be matches. This means that the last 15% or so of actual matches may be difficult to predict. These pairs of accounts behave very differently from each other, yet share a common PGP key. There could be several reasons for this. One possibility is that a seller opens an account on a different marketplace to reserve the handle, and does not end up using this account much. We observe this anecdotally, and investigate this further in Figures 2(b) and 2(c). Figure 2(b) only plots pairs where both accounts are in the top 30% of sales volumes, eliminating dormant accounts as described—this corresponds to accounts exceeding roughly $11,000 in sales. Figure 2(c) weights each pair by the smaller of the sales volumes in the pair, hence down-weighting pairs involving dormant accounts. Both these plots show marked improvements in recall.

Dormant accounts are not the only reason for false negatives. Some other possibilities are impersonators who copy a vendor's PGP key, meaning that the heuristic labels used for evaluation (and training) are incorrect. On the other hand, pairs incorrectly labeled as non-matches (i.e., same sellers posting different keys) could also affect classifier performance, in the sense that the entire range of behaviors associated with a pair of accounts belonging to the same seller are not captured by the model. We will investigate this further in the case studies described below.

Turning to false positives, through manual examination, we see that many of these accounts actually belong to the same seller, although they posted different PGP keys. These sellers might have used different marketplaces in non-overlapping time periods, even years apart, and their PGP keys might have expired or they might have lost their private keys. More can be done in terms of quantifying precisely the extent of this problem, and this is elaborated upon in Section 7.

As a secondary evaluation, Figure 3 shows the same precision-recall plot with respect to Labeled set 3. As described in Section 5, Labeled set 3 is the union of PGP labels in Labeled set 1, and Grams labels (Definition 4.3), including transitive closures. This set of labels involves pairs from 18,023 accounts, compared to the earlier 12,121, and involves additional PGP matches (from later in the data collection period), as well as labels generated in a different way from what was used to train the model (using additional Grams data as well as transitive closures). This results in a pessimistic estimate of the generalization error of the classifier. Figure 3 shows that precision is not much poorer, but recall does suffer. The false negatives issue that was described earlier is exacerbated by the additional links reported by Grams.

6.2 Clustering accuracy

Next, we re-evaluate performance after the clustering step, using the four types of linkages as described in Section 5. We use the classifier trained on PGP labels as described in Section 5, sampling 10 million non-matches (since from Figure 2(a) this produces the best performance). Predictions are generated on all pairs, including unlabeled pairs, and we then run hierarchical clustering using dissimilarity cutoffs at regularly spaced intervals from 0 to 1. We compute the precision and recall at each, evaluated using pairs with non-missing labels. The results are in Figure 4. Minimax and average linkage have superior performance, but minimax linkage has the further advantage of interpretability (see Section 5). For this curve, the bend occurs at around a cutoff of 0.74, optimizing the trade-off between precision and recall. Both of these are around 0.8 with this cutoff.

The final choice of cutoff and/or linkage method depends on the type of performance desired. For example, if an individual might be implicated in a crime, false positives could be extremely undesirable, in which case we want a very high level of precision. If we are interested in generating investigative leads for a particular account, and will be reviewing matches manually, we might instead prefer high recall. In this case, transitive closures may not be a concern either, and we might simply select pairs for which the classifier produces higher predictions, for manual review.

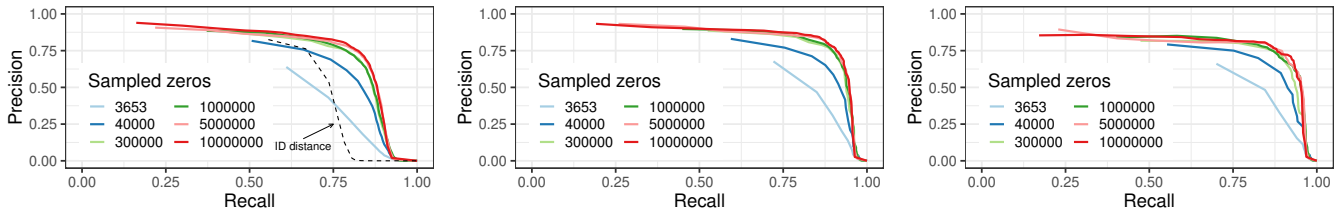
As a final step, we use the results from the clustering step above, using minimax linkage with a cutoff of 0.74. These modeling choices result in assigning the 22,163 accounts to 15,652 distinct sellers. 12,155 of the sellers are singletons, and the remainder operate multiple accounts. For instance 1,909 sellers operate 2 accounts; at the opposite end of the spectrum, 2 users operate 11 accounts. More interestingly, hundreds of sellers operate four or more accounts, which validates the motivation for this work.

6.3 Case studies

We next look at a series of case studies, combining publicly available criminal complaints with online discussions on accounts of interest.

6.3.1 Court records. As noted earlier, ground truth regarding online anonymous markets is elusive. However, when an individual is arrested for allegedly selling goods on an online anonymous marketplace, the court usually lists (some of) the various aliases under which they operated. To assess how well our algorithm could independently infer documented matches, we manually looked at

Figure 2: Precision vs. recall varying the number of non-matches sampled, using 50 trees. The left hand plot shows results for all accounts; the middle plot only considers the top 30% of accounts in sales volume (i.e., those who have sold more than $11,617.81 worth of product); the right plot weighs each point by the account's sales volume.



(a) All seller accounts. **(b)** Top 30% of seller accounts by volume. **(c)** All seller accounts (weighted by volume).

Figure 3: Precision vs. recall using different labeled sets.

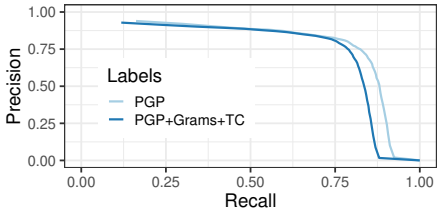
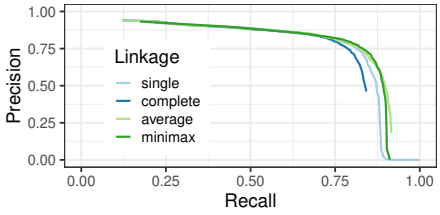


Figure 4: Precision vs. recall after hierarchical clustering.



criminal complaints, indictments, and sentencing statements culled from multiple sources (press articles, DoJ releases, etc.) corresponding to 195 distinct individuals. These documents report on 130 different screen names—some of which may have been used on multiple marketplaces—grouped in 103 families. There is no fixed reporting rule in these documents, and different accounts usually need to be significantly different to be mentioned separately; mere case changes are unlikely to be mentioned. As a result, infering account clusters using these documents is not straightforward.

With these caveats in mind, we infer that in 12 cases, a given individual is allegedly using more than one account, and in four of those, the complaint matches more than two different accounts to the same person. Out of these 12 cases, five cases mention accounts that do not appear in our database—perhaps because that account did not receive any feedback, or due to incompleteness of our data. For the remaining seven cases, we observe that, depending on the parameter specifications used, some of these are matched.

With the very conservative cut-off of 0.74 we used earlier (designed to minimize false positives), and using minimax linkage, we do not find any of these seven matches; reducing the cutoff slightly to 0.5 and still using minimax linkage, we find the right match for one of these cases. As described in Section 6, when an automated algorithm is used to generate leads for manual review, as expected to be the case in a criminal investigation, we might choose parameter specifications designed to produce high recall instead. Further, resolving transitive closures is less important. We hence look directly at the pairwise results from the random forest classifier, sampling 10 million non-matches (this corresponds to the red curve in Figure 2(a)). Models sampling a smaller number of non-matches produce predictions that are even less conservative, but we do not discuss that in detail here. Examining these pairs for the remaining six cases, we find that in half of them, at least one alternate account is matched with prediction 0.2 or higher.[4] In the remaining three cases, sellers did such an excellent job compartmentalizing their businesses over multiple accounts, for example selling different categories of products on different accounts, that the algorithm was unable to find matches in those cases.

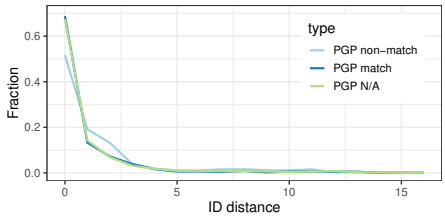
6.3.2 Adversarial examples. As described, we have designed a system to detect some subset of adversaries. We look at two scenarios of interest in more detail, and examine where the algorithm succeeds and fails: (1) accounts belonging to the same vendor, but having different screen names and PGP keys; and (2) impersonators copying a screen name and/or PGP key.

In the first scenario, we found a large number of accounts with different screen names and PGP keys that cluster, even using a conservative cutoff of 0.74 and minimax linkage. At a pairwise level, this resulted in 12,320 pairs being predicted to be matches. Of these, 2,910 had common PGP keys, 757 did not have common keys, and the remaining 8,653 pairs had a missing label, meaning that one or both accounts in the pair did not post a key. The distribution of edit distances of IDs for these predicted matches is in Figure 5. Over 30% of pairs had different IDs. Broken down by labeled match status (using training labels), pairs posting different PGP keys tend to have larger differences in screen names, more strongly suggesting adversarial intent.

Additionally, we verified some of these predicted matches manually; some do not attempt to conceal their identity, providing this information in their profile descriptions, while others do not explicitly do so. We attempted to confirm such cases using online discussion forums such as Reddit. Some examples are `FTB` on Black Market Reloaded being matched to `fredthebaker` on other marketplaces, `kaypee911` on Black Market Reloaded being matched to `aaapee911`, `evopee911`, etc. on other marketplaces, and `Sheld0nC00per` on Pandora being matched to `KeithLemon` elsewhere. All these matches despite not having a common PGP key in the training data.

[4] While this might seem like a low threshold, it represents less than 0.01% of all pairs.

Figure 5: Edit distance of IDs for model predicted matches. N = 757, 2910, and 8653, respectively.



As for impersonators, we found several examples of accounts with the same screen name or the PGP key of a different user, that our algorithm successfully placed in different clusters. For instance, `LowLands` on Silk Road 2 tried to impersonate `LowLands` on other marketplaces, which was confirmed in the account profile. (*"ATTENTION !!! ON SILK ROAD 2 THERE IS A SELLER LOWLANDS CLAIMING TO BE US....THIS IS NOT TRUE!!!..BE AWARE !!!!!!"*).

Similarly `gotmilkreplica` copied `gotmilk`'s PGP key, and posted on forums claiming to be `gotmilk`.[5] The former ships knockoffs from Hong Kong, while the latter is a large seller shipping prescription medication from India, and it seems unlikely that they are in fact the same seller.

6.3.3 Model and label disagreements. Finally, we examine some examples where the model and label strongly disagree. To be specific, we notice that in Figure 3, there is a kink in the bottom right where even using the smallest cutoff for matches, the model is unable to correctly predict all matches. We discussed this in Section 6.1, noting that the problem is worse when evaluating pairs using Labeled Set 3, where the model is unable to predict some close to 10% of labeled matches. Restricting to pairs where both accounts are in the top 30% of sales volumes (as in Figure 2), there are 117 pairs which are labeled matches (according to Labeled Set 3), but have model predictions of zero.

Looking at these pairs manually, we found that 26 pairs (22% of the 117 pairs) involve a cluster of accounts on the Dream marketplace (`cannab1z`, `GlazzyEyez`, `ibulk`, `MarcoPolo420`, `MissJessica` and `mushroomgirl`), and some of `MissJessica`'s accounts on other marketplaces. Further investigation revealed that during the seizure of Hansa and AlphaBay marketplaces in 2017, the Dutch National Police gained control of at least a dozen accounts on Dream, and posted their PGP key on all of the account pages.[6] Many of the accounts in this cluster were victims of this takeover. This case study highlights the problems with solely using PGP keys or Grams labels for matching, and suggests that the error rates reported when evaluating our model against these labels are an overestimate.

7 DISCUSSION

Classifier performance. Looking at variable (or feature) importances from the random forest classifier, we can better identify which features specifically are important for an adversary to stage a successful attack. Using the classifier sampling 10 million non-matches, we present variable importances in Table 2, measured using mean decrease in Gini impurity.[7] The items sold through each account, and their associated information plays a large role in determining if a pair of accounts is a match or not. The implication is that for an impersonation attack to succeed, an impersonator would have to sell products with item titles and descriptions very similar to those in the account that they are impersonating. This is hard to forge, as sales actually need to be confirmed by feedback for our classifier to consider the associated accounts.

Variable	Mean Gini decrease
Edit distance between IDs	3690
Jaccard similarity between item title tokens	1109
Jaccard similarity between item description tokens	697
Jaccard similarity between profile tokens	206
Same or different marketplace	160
Difference between number of item title tokens	109
Difference between number of item description tokens	93
Difference between fraction of daily sales	89
Difference between mean item price sold	88
Hamming distance between sales dates	88

Table 2: Variable importance in random forest classifier. Top 10 pairwise comparison features and their associated importance measured using mean decrease in Gini impurity.

Limitations. As described, the labels used to train the model are heuristics rather than ground truth labels. There are many inaccuracies due to the same vendor using multiple keys, or different users using the same key. In Section 6.3.3 we discussed an incident where the Dutch National Police posted the same key on multiple accounts. We have attempted to generate labels in alternative ways, and future work would involve training the model on these and comparing the results. As discussed in Section 4, generating labels from profile information using regular expression matching was not particularly successful, but this could be improved either by manual extraction or using more sophisticated tools to extract semantic from profile descriptions.

Related to this, it is difficult to determine the proportion of false positives that are actually true positives, or false negatives that are actually true negatives. As a reviewer suggested, one option would be to randomly sample cases bucketed by model score for manual review. We did this to some extent in Section 6.3.3, noting that out of 117 selected false negative pairs, at least 22% are true negatives, but a more comprehensive analysis could be done.

The algorithm itself is susceptible to adversaries that take great lengths to mimic behavior, as illustrated by several examples discussed earlier. The current methodology also assumes that account ownership does not change over time, which we know anecdotally to be false, for example due to sales of accounts or police takeovers.

Finally, scalability is a notable limitation, since our current implementation is $O(n^2)$. Generating some of the pairwise features is extremely slow and memory-intensive. Likewise, training the model is very memory-intensive. It was prohibitively expensive to sample more than 10 million non-matches.

[5] https://bitcointalk.org/index.php?topic=834362.0

[6] https://www.bleepingcomputer.com/news/security/crooks-reused-passwords-on-the-dark-web-so-dutch-police-hijacked-their-accounts/.

[7] The Gini impurity decreases after each split. For a single tree, summing these whenever a particular feature is used in the split gives the decrease in Gini impurity for that feature. Taking the mean over all trees gives the mean decrease in Gini impurity, and provides a measure of feature importance.

8 CONCLUSION

Building on an eight-year data collection effort, we implemented various methods (classification and clustering) to infer relationships between disparate vendor accounts on dark net marketplaces. We have shown that it is possible to develop a classifier to match these accounts in an adversarial context, where sellers may not want their accounts to be linked, or may attempt to impersonate others.

In aggregate, our classifier performs reasonably well (precision and recall both above 75%) both on unseen test data, as well as data labeled differently from how the model was trained. The classifier performs particularly well (recall close to 90%) for accounts with significant (> $11,000) sales volumes. Because ground truth is not generally available, actual performance may even be higher. Anecdotal evidence from manually investigating false positives and false negatives suggests mislabeled entries exist in the test set.

Additionally, we could confirm examples of correct matches and non-matches in situations where partial ground truth was available (e.g., publicly available criminal complaints). There remains room for improvement, particularly in terms of scalability and inferring more accurate labels for both training and testing.

Matching seller accounts on anonymous marketplaces gives researchers a better understanding of the ecosystem, by providing more accurate figures on the number of unique sellers involved: while the vast majority of sellers only operate one account, some sellers may be behind as many as 11 accounts. Finally, we hope that our work gives insights into solving similar adversarial matching problems in other domains. In particular, studying tradeoffs between performance and the selection features resilient to adversarial tampering is key.

9 ACKNOWLEDGMENTS

We thank Jeremy Thomas for maintaining and further developing the scraping infrastructure, Emily Rah for collecting and analyzing dark net market related criminal complaints, and Behtash Banihashemi for many discussions during the course of this research. We also thank James Arps, Mahmood Sharif, and Zachary Weinberg, for extensive feedback on earlier revisions of this manuscript. This research was partially supported by DHS Office of Science and Technology under agreement number FA8750-17-2-0188; and by the Center for Statistics and Applications in Forensic Evidence (CSAFE) through NIST Cooperative Agreement #70NANB15H176.

REFERENCES

[1] P. Agten, W. Joosen, F. Piessens, and N. Nikiforakis. 2015. Seven months' worth of mistakes: A longitudinal study of typosquatting abuse. In *Proc. ISOC NDSS*.
[2] C. Aliens. 2017. The Darknet Search Engine 'Grams' is Shutting Down. https://web.archive.org/web/20180124070700/https://www.deepdotweb.com/2017/12/15/darknet-search-engine-grams-shutting/. Accessed May 18, 2019.
[3] Anonymous. 2017. Grams: Search the Darknet. Was at http://grams7enufi7jmdl.onion. Taken offline in December 2017.
[4] J. Bien and R. Tibshirani. 2011. Hierarchical Clustering With Prototypes via Minimax Linkage. *J. Am. Stat. Assoc.* 106 495 (2011), 1075–1084.
[5] J. Broséus, D. Rhumorbarbe, C. Mireault, V. Ouellette, F. Crispino, and D. Décary-Hétu. 2016. Studying illicit drug trafficking on Darknet markets: Structure and organisation from a Canadian perspective. *Forensic Sci. Int.* 264 (2016), 7–14.
[6] P. Christen. 2012. *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection.* Springer.
[7] N. Christin. 2013. Traveling the Silk Road: A measurement analysis of a large anonymous online marketplace. In *Proc. WWW'13*. Rio de Janeiro, Brazil, 213–224.
[8] N. Christin. 2017. An EU-focused analysis of drug supply on the AlphaBay marketplace. EMCDDA report for contract CT.17.SAT.0063.1.0. Available at http://www.emcdda.europa.eu/system/files/attachments/6622/AlphaBay-final-paper.pdf.
[9] DHS S&T – CSD. [n. d.]. Information Marketplace for Policy and Analysis of Cyber-risk & Trust (IMPACT). Retrieved May 18, 2019, from https://impactcybertrust.org.
[10] R. Dingledine, N. Mathewson, and P. Syverson. 2004. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium*.
[11] M. Dittus, J. Wright, and M. Graham. 2018. Platform criminalism: the last-mile geography of the darknet market supply chain. In *Proc. of the 2018 Web Conference*. Lyon, France, 277–286.
[12] J. Douceur. 2002. The Sybil Attack. In *Proc. IPTPS '02*. Cambridge, MA.
[13] I. Fellegi and A. Sunter. 1969. A Theory for Record Linkage. *J. Am. Stat. Assoc.* 64, 328 (1969), 1183–1210.
[14] M. Gilbert and N. Dasgupta. 2017. Silicon to syringe: Cryptomarkets and disruptive innovation in opioid supply chains. *Int. J. Drug Policy* 46 (2017), 160–167.
[15] G. Kappos, H. Yousaf, M. Maller, and S. Meiklejohn. 2018. An Empirical Analysis of Anonymity in Zcash. In *Proc. USENIX Security*.
[16] P. Kintis, N. Miramirkhani, C. Lever, Y. Chen, R. Romero-Gomez, N. Pitropakis, N. Nikiforakis, and M. Antonakakis. 2017. Hiding in plain sight: a longitudinal study of combosquatting abuse. In *Proc. ACM CCS*. 569–586.
[17] S. Kumar, J. Cheng, J. Leskovec, and V.S. Subrahmanian. 2017. An Army of Me: Sockpuppets in Online Discussion Communities. In *Proc. WWW*. Perth, Australia, 857–866.
[18] J. Martin. 2014. *Drugs on the dark net: How cryptomarkets are transforming the global trade in illicit drugs.* Springer.
[19] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. Voelker, and S. Savage. 2013. A fistful of bitcoins: characterizing payments among men with no names. In *Proc. ACM/USENIX IMC*. Barcelona, Spain, 127–140.
[20] T. Moore and B. Edelman. 2010. Measuring the Perpetrators and Funders of Typosquatting. In *Proc. IFCA Financial Crypto*. 175–191.
[21] M. Möser, K. Soska, E. Heilman, K. Lee, H. Heffan, S. Srivastava, K. Hogan, J. Hennessey, A. Miller, A. Narayanan, and N. Christin. 2018. An Empirical Analysis of Traceability in the Monero Blockchain. In *Proc. PETS*, Vol. 3. Barcelona, Spain.
[22] L. Norbutas. 2018. Offline constraints in online drug marketplaces: An exploratory analysis of a cryptomarket trade network. *Int. J. Drug Policy* 56 (2018), 92–100.
[23] N. Popper. 2015. The tax sleuth who took down a drug lord. https://www.nytimes.com/2015/12/27/business/dealbook/the-unsung-tax-agent-who-put-a-face-on-the-silk-road.html. Last accessed: May 18, 2019..
[24] R. R. Sokal and C. D. Michener. 1958. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin* 38 (1958), 1409–1438.
[25] K. Soska and N. Christin. 2015. Measuring the Longitudinal Evolution of the Online Anonymous Marketplace Ecosystem. In *Proc. USENIX Security*. Washington, DC, 33–48.
[26] J. Szurdi, B. Kocso, G. Cseh, J. Spring, M. Felegyhazi, and C. Kanich. 2014. The Long "Taile" of Typosquatting Domain Names. In *Proc. USENIX Security*. San Diego, CA, 191–206.
[27] The Grugq. 2017. Operational Security and the Real World. https://medium.com/@thegrugq/operational-security-and-the-real-world-3c07e7eeb2e8. Retrieved May 18, 2019.
[28] United States District Court, Eastern District of California. 2016. Affidavit of Matthew Larsen. https://www.justice.gov/usao-edca/file/836576/download, accessed 2017-08-20.
[29] United States District Court, Eastern District of New York. 2016. Affidavit in Support of Removal to the Eastern District of California. https://regmedia.co.uk/2016/08/12/almashwali_arrest.pdf, accessed 2017-08-20. dark51.
[30] R. van Wegberg, S. Tajalizadehkhoob, K. Soska, U. Akyazi, C. Hernandez Ganan, B. Klievink, N. Christin, and M. van Eeten. 2018. Plug and Prey? Measuring the Commoditization of Cybercrime via Online Anonymous Markets. In *Proc. USENIX Security*. Baltimore, MD.
[31] S. Ventura, R. Nugent, and E. Fuchs. 2015. Seeing the non-stars: (Some) sources of bias in past disambiguation approaches and a new public tool leveraging labeled records. *Research Policy* 44, 9 (2015), 1672–1701.
[32] B. Viswanath, M. Bashir, M. Zafar, S. Bouget, S. Guha, K. Gummadi, A. Kate, and A. Mislove. 2015. Strength in numbers: Robust tamper detection in crowd computations. In *Proc. ACM COSN*. 113–124.
[33] B. Viswanath, A. Post, K. Gummadi, and A. Mislove. 2011. An analysis of social network-based sybil defenses. *ACM SIGCOMM CCR* 41, 4 (2011), 363–374.
[34] X. Wang, P. Peng, C. Wang, and G. Wang. 2018. You Are Your Photographs: Detecting Multiple Identities of Vendors in the Darknet Marketplaces. In *Proc. ACM ACIACCS*. 431–442.
[35] W. Winkler. 2006. *Overview of record linkage and current research directions.* Technical Report Statistics #2006-02. Bureau of the Census.

A APPENDIX: REPRODUCIBILITY

Our scraped and parsed marketplace data is stored in a set of SQL databases which total approximately 60 GB. Part of this was used in previous studies [7, 25] and is currently available through the IMPACT portal [9]. The remainder (data from Berlusconi, Traderoute, Valhalla, and Dream) is available through the project Github. We provide in this paper a link to the publicly available Grams data, as well. Except for Grams, these more recent datasets will be shortly available on IMPACT as well. Note that somebody interesting in reproducing the whole study from scratch would have to obtain the *non*-anonymized versions of the market data, as many features require for instance full item descriptions.

From these pre-processed source datasets, we use a Python (Python 3.4.3) script to extract PGP keys from vendor profiles and item descriptions via a number of hand-tuned heuristics. The script utilizes SQLite3 bindings for Python3 but no other non-standard libraries. The vendor key mappings are then aggregated and merged with Grams data to produce all of the vendor relations from Section 4. This script requires 22 GB of RAM and takes approximately 1 hour to run on an Intel 6700-K processor where the majority of time is occupied in reading the large input datasets as well as the single-threaded computation time of parsing PGP keys and computing the transitive closure of the relations.

Additionally we use a Perl script that interacts with the GPG command-line utility (GnuPG 2.2.12 / libgcrypt 1.8.4) to build a keyring from PGP keys that have been extracted. This step allows us to identify which keys that have been used to sign the public keys extracted from user profiles and item descriptions. The relationships between keys extracted from this process are then synthesized as a SQLite3 database that is then provided to compute the relation from definition 4.2. This step takes approximately 3 hours to run, requires very low RAM and is bound by single-threaded CPU performance.

In the last step, the processed data (source databases and data labels) is ingested by R code where we extract account-level information as well as train and evaluate the model. Training the largest model requires about 70GB of memory and the entire process takes roughly 30 hours to complete, however this is not a fundamental limitation of our approach but rather an artifact of the implementation.

All code and data for this project is available at https://github.com/xhtai/heisenbrgr/ and includes installation and setup instructions. This code has been tested against free open-sourced Linux builds and does not require any proprietary packages.