Watch Your Back: Identifying Cybercrime Financial Relationships in Bitcoin through Back-and-Forth Exploration

Gibran Gomez
IMDEA Software Institute
Universidad Politécnica de Madrid
gibran.gomez@imdea.org

Pedro Moreno-Sanchez
IMDEA Software Institute
Madrid, Spain
pedro.moreno@imdea.org

Juan Caballero
IMDEA Software Institute
Madrid, Spain
juan.caballero@imdea.org

ABSTRACT

Cybercriminals often leverage Bitcoin for their illicit activities. In this work, we propose back-and-forth exploration, a novel automated Bitcoin transaction tracing technique to identify cybercrime financial relationships. Given seed addresses belonging to a cybercrime campaign, it outputs a transaction graph, and identifies paths corresponding to relationships between the campaign under study and external services and other cybercrime campaigns. Back-and-forth exploration provides two key contributions. First, it explores both forward and backwards, instead of only forward as done by prior work, enabling the discovery of relationships that cannot be found by only exploring forward (e.g., deposits from clients of a mixer). Second, it prevents graph explosion by combining a tagging database with a machine learning classifier for identifying addresses belonging to exchanges.

We evaluate back-and-forth exploration on 30 malware families. We build oracles for 4 families using Bitcoin for C&C and use them to demonstrate that back-and-forth exploration identifies 13 C&C signaling addresses missed by prior work, 8 of which are fundamentally missed by forward-only explorations. Our approach uncovers a wealth of services used by the malware including 44 exchanges, 11 gambling sites, 5 payment service providers, 4 underground markets, 4 mining pools, and 2 mixers. In 4 families, the relations include new attribution points missed by forward-only explorations. It also identifies relationships between the malware families and other cybercrime campaigns, highlighting how some malware operators participate in a variety of cybercriminal activities.

CCS CONCEPTS

• **Security and privacy-Malware and its mitigation**;

KEYWORDS

Cybercrime Financial Relations, Blockchain, Malware, Clipper

ACM Reference Format:

Gibran Gomez, Pedro Moreno-Sanchez, and Juan Caballero. 2022. Watch Your Back: Identifying Cybercrime Financial Relationships in Bitcoin through Back-and-Forth Exploration. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22), November 7–11, 2022, Los Angeles, CA, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3548606.3560587

1 INTRODUCTION

Cryptocurrencies such as Bitcoin are attractive for cybercriminals due to their decentralized nature, (pseudo-)anonymity, irreversible transactions, and the ease to buy and sell them. However, some cryptocurrencies such as Bitcoin have a public transaction ledger, which has enabled the analysis of diverse cybercrime activities such as ransomware [28, 39, 47, 57, 59, 65], thefts [53], scams [21, 53, 58], human trafficking [60], hidden marketplaces [25, 46, 62], money laundering [54], and cryptojacking [67].

A key component of a Bitcoin cybercrime investigation is the analysis of how money flows, e.g., how cybercriminals finance their activities and where the profits of those activities go. Identifying the financial relationships (relations for short) between a cybercrime campaign and external services (e.g., exchanges, mixers, underground markets), or with other cybercriminal campaigns, is fundamental in attribution. For example, if a coin flow is identified between a malicious campaign of interest and a service (e.g., an exchange) with know-your-customer (KYC) requirements [32], the service can be leveraged as an attribution point by law enforcement to identify the perpetrators. Moreover, identifying relations between cybercrime campaigns, which were initially considered independent, sheds light into services supporting cybercriminal activities [26], and their operators, which become then targets for takedown [29, 31, 69]. Yet another application is identifying the malicious clients of a specific service, e.g., a mixer suspected of laundering money from illicit activities. This information has been fundamental in past takedown actions, by justifying to a judge that the service is a hotspot of malicious activity. For example, the arrest warrant for the *Bitcoin Fog* mixer operator was justified on its abuse for money laundering by multiple underground markets [22].

A common technique for analyzing Bitcoin abuse is *forward transaction tracing*, i.e., tracing the flow of BTCs starting from a given set of *seed addresses* known to belong to a cybercrime campaign [39, 46, 57, 58]. Examples of seed addresses are those in ransom notes (e.g., [39, 57]) and those in scam emails (e.g., [58]). But, forward transaction tracing has two fundamental limitations. First, it only considers forward propagation of the funds from the

CCS '22, November 7–11, 2022, Los Angeles, CA, USA.

ACM ISBN 978-1-4503-9450-5/22/11…$15.00
https://doi.org/10.1145/3548606.3560587

seeds, i.e., it does not explore backwards. For example, if the seed is used as input to a transaction with two inputs and one output, it traces the output, but it does not trace (backwards) the other input, which we can also attribute to the campaign since multiple inputs of the same transaction are associated to the same actor [53]. Second, if the output address receives another deposit transaction (where the seed is not an input) that deposit transaction is ignored. Thus, forward-only exploration may often miss fundamental relations. For instance, malware that uses the Bitcoin blockchain for signaling C&C servers (e.g., [59, 66]) needs to seed the signaling addresses with funds. Failure to explore backwards from the signaling addresses may miss the exchange used to seed the C&C signaling that can be leveraged as an attribution point. Furthermore, since the funds are consumed to pay the fees of the signaling transactions, there may not be a cash out that forward-only exploration can identify. Another situation is when examining the clients of a potentially malicious service, e.g., a mixer. Forward-only exploration from the mixer addresses is useless as the mixer's goal is precisely to prevent forward tracing. However, backwards exploration from the mixer's addresses allows identifying the clients that leverage the mixer, which can be used to justify an arrest warrant [22]. Last but not least, existing approaches perform shallow explorations to prevent the transaction graph from exploding in size. We posit that the key challenge to prevent such explosion, and also for identifying relations, is to identify *change of ownership*, i.e., when BTCs change hands from the operators of the campaign under study to other entities. Prior works detect change of ownership using commercial databases of tagged addresses [13, 14]. However, such databases have limited coverage, which can make the graph explode. For example, failing to identify that the cybercriminals cashed out at an exchange would include a massive amount of transactions from other innocent clients of the exchange into the transaction graph.

To address these limitations, we contribute the following:

- We propose *back-and-forth exploration*, a novel automated transaction tracing technique to identify cybercrime financial relationships. Given a set of seed addresses belonging to a cybercrime campaign, it outputs a transaction graph and it extracts paths in the transaction graph that capture relations to other entities and campaigns. The transaction graph can be used as evidence of the relations identified, and can be shared with law enforcement and other analysts for independent validation, or with a judge to justify a warrant. Moreover, the relations could be publicly shared, preventing services (e.g., exchanges) that interact with the cybercrime campaigns from claiming that they were unaware of the abuse. Back-and-forth exploration provides two key contributions over current forward transaction tracing techniques. First, it explores both forward and backwards, enabling the discovery of important relations missed otherwise. Second, it combines the use of a tagging database with a machine learning classifier to identify addresses belonging to exchanges. The classifier can detect previously unknown (i.e., untagged) exchange addresses, preventing graph explosion.
- We propose *operation oracles*, binary classifiers that can identify specific types of malicious addresses using sequences of distinctive transactions. We build operation oracles for four malware families that use the Bitcoin blockchain for signaling C&C servers. We use them to identify signaling addresses in the transaction graphs, demonstrating that back-and-forth exploration can discover 13 signaling addresses missed by previous works [40, 59, 66] and that backwards exploration is key for their discovery.
- We build a database of 230K tagged addresses from open sources and expand it into 116M tagged addresses through multi-input clustering. We show that tag databases need to handle *double ownership*, i.e., Bitcoin addresses with more than one owner. Double ownership is common in services offering online wallets, which create an address for a specific user. In that scenario the service owns the address (i.e., has the private key), but the address is handled (and thus indirectly owned) by the user for which it is created.
- We apply back-and-forth exploration on 30 malware families: 19 clippers that replace addresses in the OS clipboard of the infected host with an address controlled by the malware operators, 7 ransomware, and 4 families using Bitcoin for C&C. Back-and-forth exploration finds relations in 93% of the explorations. For 13% of the families, it identifies new attribution points, missed by forward-only explorations. It uncovers that 23 replacement addresses for 9 clippers are online wallets in exchanges (double ownership). This shows that clippers, a largely overlooked, but highly effective, malware class, often directly cash out the stolen funds without intermediate steps. It is surprising that the abusive nature of those addresses goes unnoticed for the exchanges. Overall, the most common relations are with 44 exchanges. Exchanges used by multiple families (e.g. *localbitcoins*, *coinpayments*, *coinbase*, *poloniex*, *cryptonator*, *btc-e*) are often not the largest ones, which could indicate a more lax approach to abuse detection. In fact, one of those exchanges has already been taken down [69]. Other identified relations are 2 mixers (*wasabi*, *bitcoinfog*), the latter being recently taken down for money laundering [29]; 11 gambling sites likely also used for money laundering; 5 payment services used for acquiring resources; 4 mining pools, and 4 underground markets. In 7 explorations, relations are found between a malware family and other cybercrime entities. We observe that some malware operators participate in a variety of cybercriminal activities. For example, the *MrPr0gr4mmer* operator sells spam tools, sends sextortion scam campaigns, and handles a clipper. We also identify clippers run by the same operators and observe malware operators using profits to upgrade their software arsenal (e.g., buying licenses for malware kits and spam tools) and for seeding new C&C signaling approaches. Finally, we perform a backwards-only exploration on the *bitcoinfog* mixer identifying 15 underground markets, 130 other services, and 15 malicious operations that leveraged it as clients.
- We release our exploration code and tag database [19]. We also provide an extended version with further details [34].

2 STATE OF THE ART

2.1 Bitcoin Address Ownership

The notion of *ownership* of a Bitcoin address is associated to the user who possesses the corresponding signing key and can thus authorize a transfer of coins from that address. A user may own multiple addresses. For example, a (privacy-aware) user may create a new address for every payment he receives.

Multi-input clustering. We leverage a popular technique to identify Bitcoin addresses belonging to the same owner called multi-input (MI) clustering [20, 53, 56, 61]. It assumes that input addresses to the same transaction have the same owner because their private keys are used together to sign the transaction. Clusters can be created transitively. If a transaction has addresses *A* and *B* as inputs, and another transaction has *B* and *C* as inputs, then *A*, *B* and *C* are all clustered together and have the same owner. One exception are CoinJoin [51] transactions where a group of users create a single transaction that simultaneously spends all their inputs into a (shuffled) list of outputs. For this reason, before computing MI clustering, we apply proposed heuristics to identify CoinJoin transactions [33, 43]. There exist other address clustering heuristics such as shadow address [20], change address [53], and its variants (peeling chain, power of ten, address reuse) [43]. However, those techniques rely on usage patterns that change over time and can lead to high false positives.

Tagging. While MI clustering identifies addresses belonging to the same owner, it does not determine who the owner is. For this, commercial services (e.g., [13, 14]) provide paid access to databases of addresses tagged with a label stating their owner or usage. Instead, we leverage public resources to build our own tag database. An intrinsic limitation of tag databases is that they only include a small fraction of Bitcoin addresses, having thus a low coverage. In this work, we show that when tagging addresses, it is fundamental to handle *double ownership*, i.e., that a Bitcoin address may have more than one owner. In the double ownership scheme, a first user (i.e., the *owner*) possesses the signing key corresponding to the address and accepts transaction requests (possibly authenticated with e.g., username and password) from a second user (i.e., the *beneficiary*) who is thereby the one governing where the coins are transferred and when. Note that the owner does not share the signing key to the beneficiary, so the beneficiary cannot produce transactions on its own. This double ownership scheme is used for instance by some exchanges and online wallets. These services, on user’s request to deposit its coins, may create a fresh address (and the corresponding signing key) where afterwards the user’s coins are deposited. Handling double ownership is fundamental when collecting tagged addresses from multiple public sources. For example, the same address may be tagged in one source as belonging to an exchange and in another source as being the replacement address for a clipper malware. Without taking double ownership into account that would create a conflict in assigning ownership to the address. Instead, the address should be tagged as belonging to an exchange (i.e., the owner) that has assigned that address to an exchange user (i.e., the beneficiary) who happens to be the clipper’s operator. Our tag database construction is detailed in Section 4.

Exchange address classifier. We propose a machine learning classifier to identify addresses belonging to exchanges, even if the specific exchange it belongs to was never tagged and is therefore not part of the training dataset. Our exchange classifier prevents graph explosion due to limited tagging coverage, but can only determine a given address belongs to an exchange, not to which specific exchange. Our exchange classifier is detailed in Section 5.

Operation oracles. Previous works build functions that given a ransomware collection address distinguish victim payments by examining if deposit amounts are within payment ranges used by the family [28, 39, 47, 57]. We expand this concept by proposing *operation oracles*, binary classifiers that identify specific types of malicious addresses (performing distinctive transactions) by applying rules to the sequence of transactions involving the address. We build operation oracles, detailed in Section 6, for three malware families that leverage the Bitcoin blockchain to signal C&C endpoints. Our operation oracles consider transaction sequences, e.g., that address A sends a payment to address B and soon after address B returns the same amount, minus the transaction fee, back to A. Considering multiple transactions and dependencies between them makes the classifier more distinctive than considering only transaction values, reducing false positives (FPs).

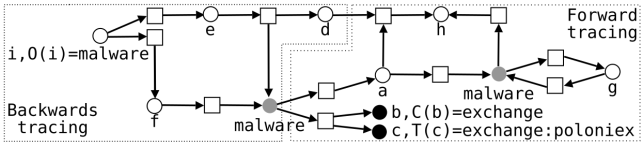

Figure 1: Example transaction graph from two malicious seeds in gray. Forward transaction tracing would discover the right subgraph, while back-and-forth exploration would discover the whole graph including a new malicious address.

2.2 Transaction Tracing

Forward transaction tracing focuses on the outgoing transactions performed by the seeds and transitively by their destination addresses [39, 46, 57, 58]. A main limitation of forward transaction tracing is that it only explores forward, but it does not explore backwards. In particular, in forward transaction tracing, if an address is being traced, then the outputs of all transactions where that address is used as input will also be traced. But, other inputs in those transactions will not be traced. In addition, deposit transactions into a traced address, where the traced address appears as output, but no input is currently traced, are not traced either. Due to this limitation, forward transaction tracing cannot discover relations only reachable by exploring backwards. For example, forward transaction tracing starting from addresses belonging to a mixer makes the exploration explode due to the mixing behavior, while exploring backwards can reveal the clients using the mixer. In another example, exploring backwards from a C&C signaling address can reveal the exchange used to fund the signaling. But, forward transaction tracing may not reveal such attribution point as the funds are consumed in the signaling.

To address this limitation we propose a novel *back-and-forth exploration*, which traces forward and backward. In a bit more detail, beginning from a seed, our back-and-forth exploration retrieves deposit addresses of incoming transactions, withdrawal addresses of incoming transactions, deposit addresses of outgoing transactions, and withdrawal addresses of outgoing transactions.

We illustrate the differences between forward transaction tracing and back-and-forth exploration using the example transaction graph in Figure 1. In the graph, circle nodes correspond to addresses and square nodes to transactions. Nodes filled in black are those belonging to a cluster tagged as exchange. Edges capture coins flowing from an input address to a transaction, or from a transaction to an output address. The exploration starts from the two seeds in gray. The graph is split in two. The right side corresponds to the graph that forward transaction tracing would produce and the left side is the additional subgraph that back-and-forth exploration would additionally discover, including a new malware address. There are two points were both explorations differ. First, back-and-forth exploration traces deposit addresses to the leftmost seed, including addresses f, e, and d into the graph. Second, once address a has been discovered through a forward step from the leftmost seed, the transaction where a is an input and h is an output will be traced. At that point, back-and-forth exploration would include address d in the tracing while forward transaction tracing would not.

One risk of backwards exploration is that any entity can deposit BTCs into any address. If a benign entity sends funds to a malicious address, the backwards exploration may consider the sender address part of the malicious campaign. Since addresses in ransom notes and scam emails receive victim's payments and are typically seeds for our back-and-forth exploration, our platform has an option to prevent exploration backwards from the seeds. Note that even if this option is used, back-and-forth exploration still explores backwards at intermediate nodes. For example, in Figure 1, even if backwards exploration on the seeds is disabled, the final graph would be the same because the transaction between a and h would bring d into the graph, and successive backwards steps would then include e, i, and f. There are two other cases where backwards exploration can be problematic. One case are *forced address reuse* (or *dust*) attacks where an entity performs micro-payments to addresses it does not own, hoping to incentivize the receivers to move those small amounts in later transactions [39]. In this work, we use some filtering heuristics to identify and remove forced address reuse transactions. The other case is researchers trying to hijack the C&C infrastructure of a malware by sending payments to the signaling addresses [66]. In such cases, researchers are likely to specify their transactions to avoid being incorrectly linked to the malicious campaign, allowing their filtering.

Change of ownership. A fundamental challenge when tracing a malicious campaign is to detect change of ownership, i.e., addresses that belong to different entities. A critical change of ownership is when BTCs flow into exchanges, e.g., cybercriminals cashing out their profits. Failure to detect an exchange address would make the graph explode by introducing thousands (or even millions) of transactions from other clients of the exchange. Detecting change of ownership is also fundamental to analyze relationships between the seed owners and external entities. In this case, failure to detect an exchange address would make it look like there exist relations between unrelated cybercriminal campaigns that simply use the same exchange. For example, in Figure 1 address b is identified as an exchange by our ML classifier, while c is identified as belonging to the *poloniex* exchange through a previously available tag. Failure to detect these addresses would have made the graph explode.

3 BACK-AND-FORTH EXPLORATION

Given seed Bitcoin addresses belonging to a cybercrime campaign, back-and-forth exploration outputs a transaction graph and identifies paths in the graph that capture relations to other entities and campaigns. As illustrated in Figure 1, nodes in the transaction graph are of two types: addresses and transactions. Edges are directed; they may connect an output slot in a transaction to the address where funds are received, or connect an address to the input slot in a transaction where funds are sent to. If the same address appears several times as input or output of the same transaction, a single edge connects address and transaction. Address nodes are reused if the same address appears in multiple transactions. Address nodes have attributes capturing their alphanumeric address string, the list of tags associated to the address (if any), the identifier of the MI cluster the address belongs to, and the address type (e.g., if detected by the exchange classifier or by the operation oracle). Transaction nodes have as attributes the transaction identifier and the total BTC amount transferred. Edges have as attributes the transaction ID, the slot number, and the BTC amount transferred in that slot.

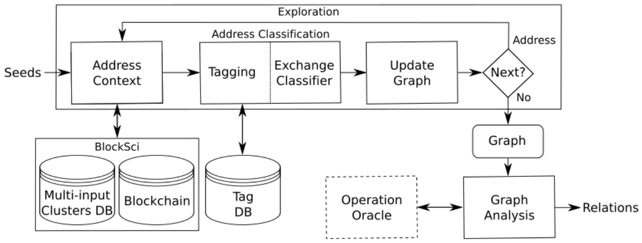

Figure 2: Back-and-forth exploration architecture.

The back-and-forth exploration architecture is shown in Figure 2. It performs an iterative process. At each iteration, it selects a Bitcoin address from the worklist and performs three steps. First, it obtains the address context by identifying the MI-cluster the address belongs to and extracting from the blockchain its deposit and withdrawal transactions. The blockchain data is processed using the BlockSci framework [43]. Before launching any exploration, BlockSci is used to parse the blockchain data up to the desired block height and to pre-compute the MI clustering results. Second, it classifies the address to check for change of ownership using the input tag database (detailed in Section 4) and the exchange classifier (detailed in Section 5). Finally, it updates the transaction graph with the obtained transactions and addresses; and adds to the worklist any new addresses that need to be explored. The process starts by placing the seeds in the worklist and finishes when the worklist is empty, or a configurable exploration limit is reached. At the end of this iterative process the transaction graph is output.

The graph analysis module takes as input the transaction graph and an optional operation oracle. It outputs relations between the campaign under study and other entities in the graph. First, it tags addresses satisfying the oracle, e.g., as a C&C signaling address. Then it identifies relations, which are paths between a seed and a tagged address that satisfy the following constraints. First, the found tag should differ from the seeds' tag. Second, the path should be directed from the seed to the tagged address, or from the tagged address to the seed, i.e., it should not mix edges with opposite directions. This constraint makes sure that the relation captures money flow from the campaign under analysis to an external entity or vice-versa. Third, addresses identified by the exchange classifier, for which no tags were available, are excluded since it is not known to what specific exchanges they belong to.

Address context. At each iteration, the address being explored is used to query BlockSci to obtain the ID of the MI cluster to which the address belongs. If the cluster has already been examined, e.g., if detected as an exchange when examining a previous address in the cluster, its classification results are recovered. Then, BlockSci is queried again to obtain the list of all transactions involving the address. CoinJoin and forced address reuse transactions are filtered from the exploration. The former are identified using a set of previously proposed heuristics [33, 43] implemented by BlockSci. The latter are identified using our own heuristic, which looks for transactions that send the same amount to a large number of (at least 100) output addresses. Next, it obtains from BlockSci the list of addresses involved in the remaining transactions, both in input and output slots. Transactions and addresses are added to the transaction graph and passed to the address classification that determines which addresses should be explored in later iterations.

Address classification. Each address received from context extraction is examined for change of ownership. Addresses identified as belonging to other entities, either through tags or by the exchange classifier, will be labeled as such and will not be traced. The rest will be added to the worklist to be traced. First, the address is queried to the tag database. If tagged, and if its tag category identifies it as a service (e.g., exchange, tormarket, gambling, etc), the address is not added to the worklist. There is no need to keep exploring the address since that would explore the service, introducing many extraneous nodes in the transaction graph. Second, the ML exchange classifier, detailed in Section 5, is applied to the address. If detected, the address is not added to the worklist. As an optimization, the ID of the address' MI-cluster is cached, so that any other address in the same cluster can be directly marked as an exchange without going through the classification. Addresses added to the worklist are assigned a priority inversely proportional to the total input and output slots in their transactions. The address with highest priority is selected and another iteration starts. The intuition behind the priority is that addresses with many transactions, or with very large transactions, could belong to services not in the tag database or to false negatives of the exchange classifier. To avoid the exploration being stuck exploring those addresses, their priority is lowered so that they are explored at the end. The priority only affects the order of the exploration as long as the exploration runs to its end. However, if the user sets a maximum exploration limit, high volume addresses may not be explored before the limit is reached and some relations may be missed.

4 TAGGING

Tag database construction comprises of two steps. First, we collect 230K addresses from public sources and tag them using their context. Then, we propagate the address tags to their MI clusters obtaining 116M tagged addresses. After each step there is a disambiguation process to resolve any conflicts due to multiple tags assigned to the same address or cluster. We explore a variety of publicly available sources to tag Bitcoin addresses including databases of already tagged addresses, (e.g., WalletExplorer tags [17] available in GraphSense [37]), web pages where services disclose their addresses (e.g., the SatoshiDice gambling service [16]), forums where users rate each other (e.g., BitcoinTalk [12], Bitcoin-OTC [11]), crowd-sourced databases where users report abusive addresses (e.g., Bitcoin Abuse [9]), and security blog articles [23]. The different sources have little overlap and each source provides some addresses that do not appear in any other source. Thus, they all contribute to making the tag database as complete as possible. Some of these sources such as WalletExplorer and Bitcoin-OTC provide their own tagging database, which can be automatically imported into our platform. For others like BitcoinTalk, tags are produced by manually inspecting specific reports. For Bitcoin Abuse, the category is provided, but selecting the label requires manually inspecting the reports. Updating the tagging database needs to be done periodically and is the only step in our approach that requires human involvement. In future work, we plan to explore automating this step using NLP.

		Addresses		Multi-Input Clustering	
Category	**Class**	**All**	**wTXES**	**Clusters**	**Addresses**
sextortion	Abuse	62,600	407	171	1,131
miscabuse	Abuse	57,031	12,018	7,568	8,656,754
mining	Service	37,454	37,453	29,526	594,609
mixuser	Individ.	36,548	36,548	7,308	36,860
ransomware	Malware	17,668	17,447	10,348	21,160
clipper	Malware	9,513	464	105	1,376
username	Individ.	6,618	5,399	5,110	153,826
exchange	Service	1,301	1,299	737	75,257,530
terrorism	Abuse	275	275	48	306
theft	Abuse	221	220	129	2,072,269
gambling	Service	182	182	106	8,800,225
onlinewallet	Service	140	129	37	4,612,732
defi	Service	93	17	17	15,945
scam	Abuse	72	61	32	314,537
mixer	Service	56	55	52	2,137,493
ponzi	Abuse	54	54	41	20,908
service	Service	50	50	43	1,293,660
malware	Malware	19	18	17	38
tormarket	Service	19	19	19	2,327,272
payment	Service	15	15	12	9,717,967
donation	Benign	14	12	8	217
state-sponsored	Abuse	14	14	8	66,855
drugtrafficking	Abuse	11	11	2	38
bankingtrojan	Malware	3	1	1	1
cryptojacking	Malware	3	3	2	25
miscbenign	Benign	3	3	1	2
webskimming	Malware	3	3	3	6
usgov	Benign	2	2	2	3
	All	229,982	112,179	61,453	116,103,745

Table 1: Tag database summary.

Each database entry is a 6-tuple (*address*, *class*, *category*, *label*, *subtype*, *urls*). There are 28 categories, detailed in Table 1, which can be grouped into 5 classes: malware, abuse, services, individuals, and benign. Label captures the ownership, i.e., the specific malware family, service, or individual that owns the address. Subtype is an optional component that details the usage of the address, e.g., if it is the hot wallet of an exchange or a malware C&C signaling address. The final component is the list of source URLs that discuss the address. For example, address *1NDyJt* has a category *exchange*, label *binance* (the specific exchange), hot wallet as subtype, and a URL to the WalletExplorer webpage that describes it. Throughout the paper, when we refer to a tag we mean its category and label, e.g., *exchange:binance*. We identify addresses by their first 6 characters, full addresses can be found in the repository [19].

We initially assign a tag to an address for each source that mentions it. Then, we select a unique address tag by applying the following criteria for resolving potential ambiguity in addresses with multiple tags. If an address has multiple tags with the same category and label, we favor the tag that provides more information, i.e., the one with a subtype, or the longest one otherwise. If an address has multiple tags with different category and label pairs, we favor tags not coming from Bitcoin Abuse because categories in that service are assigned by users with varying expertise and it is not rare that they are wrong. If an address has multiple tags of the mining category, we merge its labels to capture all the mining pools where the miner has participated. In any other cases, we manually resolve the inconsistency. If we cannot confidently assign a tag, we remove the address from the database. When applying this procedure to the 230K addresses, only 66 (0.03%) required manual inspection and we could disambiguate those, showing that this strategy is effective.

Next, we obtain the MI clusters for all the tagged addresses. If a single tagged address appears in the cluster, we assign that tag to all other addresses in the cluster. If multiple tagged addressed appear in the same cluster, we apply the following disambiguation process. If a cluster has only one service tag, we assume this is a case of double ownership where the owner of the address is the service, and the beneficiary is the owner of the other tags. If multiple service tags are present, we try to manually disambiguate them, e.g., checking if the two services are related to each other. For cases where no service tags are present, we compute the edit distance between their labels. If they are close, e.g., user:Metabank.ru and user:Metabank, we consider them aliases and merge them. Otherwise, we manually try to disambiguate them. If manual disambiguation fails, we do not expand the tags to the cluster.

Applying MI clustering to the 230K tagged addresses produces 61K clusters comprising of 116M addresses. Of the 61K clusters, 817 (1.15%) have more than one tag, from which 368 have only miscabuse tags from Bitcoin Abuse and 286 can be automatically solved since they contain multiple tagged addresses of the same owner. Of the remaining 163 clusters, we could manually resolve conflicts in 91 corresponding to 54 services with double-ownership (e.g. exchanges) and 37 equivalent tags (e.g., usernames with two aliases). We could not resolve 72 clusters, for which we do not propagate tags. Of those, 66 were clusters without service tags (63 of them with multiple Bitcoin-OTC user tags) and 6 clusters with more than one service tag (4 of them with multiple mining tags). An interesting case is one of the clusters with multiple service tags. It is due to the special address 3J98t1, which corresponds to the empty string private key. Surprisingly, that address occasionally receives deposits possibly because it appears in many Bitcoin tutorials. Any Bitcoin user can spend funds from this address (since the private key is not really secret), and many users actually do, creating a MI cluster with 13M addresses. In summary, only 0.27% of the 61K clusters required manual inspection, and we could expand tags for 99.3% of all tagged clusters, showing that our strategy is effective.

5 EXCHANGE DETECTOR

We build a binary ML classifier to identify if an address belongs to an exchange in three steps. First, we build a dataset of 108K labeled addresses. Then, we propose 42 features and use them to train different models. Finally, we evaluate the accuracy of the models.

Dataset. To train and test the classifier we build a dataset of 108,530 labeled addresses, summarized in Table 2. Half of them (positive class) are addresses tagged as exchanges by WalletExplorer [17]. The other half (negative class) are non-exchange addresses belonging to individuals, ransomware, and gambling sites. Individuals correspond to Bitcoin-OTC [11] users with a registered Bitcoin address and GPG key, and having received ratings. To avoid abusive users we require that they have received less than 100 negative ratings and have an overall positive rating. Ransomware addresses come from [57] and gambling addresses belong to SatoshiDice [16]. We obtain the MI clusters of those starting addresses and select additional addresses from the clusters until the dataset is balanced. We use 80% of the dataset for training and the other 20% for testing.

Class	Source	StAddr.	Entities	MI Clust.	Addr.
positive	WalletExplorer	167	117	151	54,265
negative	Bitcoin-OTC	923	923	788	38,702
negative	SatoshiDice	40	1	13	1,154
negative	Ransomware	7,223	65	78	14,409

Table 2: Dataset of 108,530 addresses used for training and testing exchange classifier.

Feature	New	Type	Class	Description
type	✓	Cat.	AD	Address type
equiv_addrs	✓	Int.	AD	Number of equivalent addresses
lifetime	✗	Int.	TS	Total seconds from first TX to last
timespan_d	✓	Int.	TS	Seconds from first deposit TX to last
timespan_w	✓	Int.	TS	Seconds from first withdrawal TX to last
activity	✗	Int.	TS	Number of days with at least one TX
activity_d	✓	Int.	TS	Number of days with at least one deposit TX
activity_w	✓	Int.	TS	Number of days with at least one withdrawal TX
idle_time	✓	Int.	TS	Number of days without TXES
daily_d_rate	✓	Float	TS	Rate of deposits per day in its lifetime (tx/day)
daily_w_rate	✓	Float	TS	Rate of withdrawals per day in its lifetime (tx/day)
yearly_d_txes	✓	Float	TS	Average number of deposit TXES per year
yearly_w_txes	✓	Float	TS	Average number of withdrawal TXES per year
balance	✓	Int.	TX	Balance to date, in satoshis
deposited	✗	Int.	TX	Total satoshis deposited/received into address
withdrawn	✗	Int.	TX	Total satoshis withdrawn/sent from address
txes	✗	Int.	TX	Total number of transactions
txes_out	✗	Int.	TX	Number of TXES with this address as output
txes_in	✗	Int.	TX	Number of TXES with this address as input
addr_as_change	✓	Float	TX	Ratio of TXES address used as input and output
outputs	✓	Int.	TX	Total output slots in which the address appear
inputs	✓	Int.	TX	Total input slots in which the address appear
utxos	✓	Int.	TX	Number of unspent transaction outputs to date
tx_size_mean	✓	Float	TX	Average total size (in bytes) of all its TXES
tx_weight_mean	✓	Float	TX	Average weight of TXES
tx_fee_mean	✓	Float	TX	Average fee paid in TXES
ins_age_mean	✓	Float	TX	Average blocks between input and spent output
coinbase	✗	Int.	TX	Number of TXES that are coinbase
coinjoin	✓	Int.	TX	Number of TXES that are coinjoin
coinjoin_out	✓	Int.	TX	Coinjoin TXES with this address as output
coinjoin_in	✓	Int.	TX	Coinjoin TXES with this address as input
tx_ratio	✗	Float	TX	Number of input TXES divided by output TXES
outs_per_tx	✓	Float	TX	Average number of output slots per TX
ins_per_tx	✓	Float	TX	Average number of input slots per TX
outs_per_out	✓	Float	TX	Average number of output slots per output TX
ins_per_out	✓	Float	TX	Average number of input slots per output TX
outs_per_in	✗	Float	TX	Average number of output slots per input TX
ins_per_in	✗	Float	TX	Average number of input slots per input TX
profit_rate	✓	Float	TX	Total satoshis deposited over lifetime
expense_rate	✓	Float	TX	Total satoshis withdrawn over lifetime
d_per_tx	✓	Float	TX	Total deposited amount per TX (satoshis/tx)
w_per_tx	✓	Float	TX	Total withdrawn amount per TX (satoshis/tx)

Table 3: Classifier features extracted per address.

Model training. We extract 42 features for an address, summarized in Table 3. Features can be grouped into three classes. The 29 transaction (TX) features capture properties of the transactions where the address appears such as number of (all/deposit/withdrawal/CoinJoin) transactions, number of transactions where the address appears as input or output, and statistics on the transferred amounts and paid fees. The 11 block timestamp (TS) features capture time properties such as address lifetime, time differences between noteworthy events (e.g., first and last deposits), and statistics on the daily

	Training			Testing		
Classifier	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
Random Forest	0.956	0.956	0.956	0.971	0.946	0.958
Gradient Boosting	0.949	0.948	0.948	0.965	0.935	0.950
Decision Tree	0.934	0.934	0.934	0.938	0.936	0.937

Table 4: Classifier results using Stratified 5-fold Cross Validation on 80% training data, and on 20% of unseen testing data.

rate of transactions. Finally, two address (AD) features capture the address type (e.g., pubkeyhash, multisig) and the number of other addresses (e.g., multisig) built from the address. Of the 42 features, 74% (31) are novel and the rest have been used in previous works [21, 35, 42, 49, 71]. Features are normalized using z-score to have zero mean and unit standard deviation. To understand the importance of the different features, we perform a feature analysis using mutual information (MI) gain. The feature analysis shows that the most important features are those that capture the high volume of transactions over time by exchanges The top 5 features and their MI gain are: yearly_d_txes (0.1626), yearly_w_txes (0.1601), activity_w (0.1475) txes_in (0.1463) and daily_w_rate (0.1437).

We train three models using the 42 features and 80% of the labeled dataset: Decision Tree, Random Forest, and Gradient Boosting. We focus on tree-based models because they are easy to interpret. The left side of Table 4 shows the precision, recall, and F1 score for each model, using a stratified 5-fold cross validation training. The results show that Random Forest achieves the best F1 score of 0.956.

Testing. We evaluate the models to make predictions on unseen data by using the 20% of the dataset reserved for testing. The results, on the right side of Table 4, show that the best model is again Random Forest with a F1 of 0.958, so we use that model throughout the explorations with the following hyper-parameters: 600 trees, maximum depth of 40, and prediction threshold of 0.5.

6 BACK-AND-FORTH EXPLORATION EVALUATION

We evaluate our platform on 30 malware families: 4 families that use Bitcoin for C&C, 19 clippers that replace addresses in the OS clipboard of the infected host with an address controlled by the malware operators, and 7 ransomware. We compare back-and-forth exploration with a baseline forward-only exploration where we turn off the backwards exploration step in our platform, but allow forward-only exploration to benefit from our exchange classifier and tag database. We use this baseline because prior works implement different versions of forward-only exploration that vary, among others, in the number of steps or addresses explored, whether MI clustering is used, and what tags or other heuristics are used to filter unrelated addresses. For each family, we run two explorations: back-and-forth and forward-only. Table 5 summarizes the back-and-forth explorations and the delta with the forward-only explorations (in parenthesis).

For clipper and ransomware families back-and-forth exploration is configured to avoid exploring backwards on the seeds since addresses depositing into the seeds belong to victims (SDD column). Note however, that back-and-forth exploration will still explore backwards after the first forward step from the seeds. The seed information is broken into three columns. *All* is the total addresses in our tag database for the family. *OW* captures how many of those seeds are identified as belonging to online wallets. Those seeds are not explored forward as that would explore the transactions of the service holding the online wallet, but they can still be explored backwards if they do not belong to a clipper or ransomware (i.e., skidmap). *Exp.* captures the number of addresses effectively explored, after removing addresses in online wallets and addresses without transactions. Some families like *clipsa* and *locky* have thousands of seeds. In the case of *clipsa*, when replacing the address in the clipboard, this clipper selects a visually-similar address from an address database, so that the replacement has a higher chance of not being noticed by the victim. Since the address database ships with the malware, researchers have published its contents [63]. However, only 375 (4%) of clipsa addresses have transactions and thus are effectively used in the exploration. Those 375 addresses collect nearly 18 BTC ($277K). Other families like n40 collect even larger BTC amounts (27.4 BTC, $187K) using a single replacement address. Thus, despite their simplicity, clippers can produce significant gains for their operators.

An important observation is that 24 seeds are online wallets whose owner is an exchange and whose beneficiary is the malware operator (or a money mule). Nearly all of those (23/24) are replacement addresses from 9 clipper families, the other belonging to the *skidmap* crypto miner. Thus, replacement addresses hardcoded in clipper samples are often hosted in exchanges. It is surprising that the abusive nature of those addresses goes unnoticed for the exchanges, as they are often mentioned in public abuse reports by the victims, as well as in analyses by security vendors. Note that this does not happen for addresses in ransom notes, possibly because of ransomware's higher visibility. Based on this observation, we discuss in Section 8 a defense that would examine public reports for malicious addresses, identify online wallets, and (publicly) report them to exchanges, so that it cannot be claimed that their abusive nature was not known.

The middle section of the table details the output graphs. For 7 clipper families all seeds are in online wallets. Thus, no exploration is performed for them and the graph contains only the seeds. For the remaining 23 families, 69% contain a single component. Each component contains at least one seed. Thus, multiple components indicate that some seeds were not connected to the rest of the seeds. The graphs for those 23 families contain an average of 2,274 addresses, with a maximum of 13.8K for *globeimposter*. Unexplored addresses are those not processed before the exploration limit was reached. The numbers in parenthesis show the fraction of addresses and transactions in the forward-only graph compared to the back-and-forth graph. On average, the back-and-forth graph is three times larger (3.6x addresses, 2.6x transactions) offering a more complete view of an operation. In Sections 6.1 and 7 we show that the back-and-forth graphs contain critical information not available in the forward-only graphs such as previously unknown C&C signaling addresses for Cerber and Pony, unreported relations between malware families (e.g. DMALocker and Slave), and new attribution points (Pony, Skidmap, MrPr0gr4mmer, Phorpiextldr).

The right section of the table shows the number of addresses tagged and those without a tag identified as exchanges by the ML classifier. It also shows the total number of relations found in the back-and-forth and forward-only (in parenthesis) explorations.

			Seeds			Graph				Classification			
Class	Operation	SDD	All	OW	Exp.	Comp.	Addr	Txes	Unexp.	Tagged	Exch.	Relat.	Time
C&C	cerber	✓	4	0	4	1	608 (73.7%)	1,050 (86.0%)	1	12	394	1 (0)	1'26"
	glupteba	✓	3	0	3	3	195 (17.4%)	167 (24.6%)	5	14	83	0 (0)	1'02"
	pony	✓	4	0	4	2	1,154 (2.3%)	4,252 (36.4%)	72	379	188	1 (0)	5'57"
	skidmap	✓	1	1	1	1	132 (0.8%)	107 (11.2%)	5	37	7	5 (1)	4'04"
Clipper	aggah	✗	1	0	1	1	54 (13.0%)	44 (18.2%)	6	7	19	1 (1)	1'04"
	androidclipper	✗	1	1	0	1	1 (-)	0 (-)	0	1	0	1 (1)	-
	azorult	✗	1	1	0	1	1 (-)	0 (-)	0	1	0	1 (1)	-
	bitcoingrabber	✗	1	0	1	1	504 (18.7%)	650 (46.9%)	32	203	116	0 (5)	3'19"
	casbaneiro	✗	1	0	1	1	125 (47.2%)	162 (59.3%)	5	48	55	4 (4)	0'51"
	clipbanker	✗	1	1	0	1	1 (-)	0 (-)	0	1	0	1 (1)	-
	clipsa	✗	9,412	0	375	25	4,655 (14.6%)	5,756 (14.6%)	120	1,339	2,471	6 (6)	25'18"
	cliptomaner	✗	1	1	0	1	1 (-)	0 (-)	0	1	0	1 (1)	-
	cryptoshuffler	✗	1	0	1	1	3,253 (15.6%)	7,744 (30.1%)	220	966	1,279	11 (12)	42'00"
	masad	✗	2	0	2	1	3,882 (12.1%)	7,389 (20.0%)	182	1,366	1,404	11 (11)	28'40"
	mekotio	✗	3	0	2	1	2,180 (6.7%)	4,031 (9.6%)	71	600	1,125	6 (17)	21'08"
	mekotion40	✗	3	0	3	3	1,153 (12.6%)	1,152 (27.0%)	30	507	338	6 (8)	24'25"
	mispadu	✗	1	1	0	1	1 (-)	0 (-)	0	1	0	1 (1)	-
	mrpr0gr4mmer	✗	2	0	2	1	837 (68.9%)	1,704 (92.1%)	19	321	350	2 (2)	2'18"
	n40	✗	1	1	0	1	1 (-)	0 (-)	0	1	0	1 (1)	-
	phorpiex-tldr	✗	15	14	1	1	203 (5.4%)	107 (51.4%)	14	109	62	3 (3)	2'16"
	phorpiex-trik	✗	8	1	6	4	282 (36.5%)	539 (35.0%)	2	19	116	5 (8)	1'14"
	predatorthethief	✗	1	0	1	1	2,702 (15.2%)	5,039 (32.3%)	157	1,100	947	8 (9)	38'22"
	protonbot	✗	2	2	0	1	2 (-)	0 (-)	0	2	0	2 (2)	-
Ransomware	cryptotorlocker2015	✗	1	0	1	1	336 (43.8%)	3,695 (57.8%)	31	103	125	11 (8)	122'02"
	cryptxxx	✗	1	0	1	1	4,015 (15.5%)	10,087 (18.2%)	43	1,607	984	13 (11)	31'29"
	dmalocker	✗	11	0	11	1	1,631 (29.6%)	4,855 (28.3%)	25	525	531	8 (6)	3'44"
	globeimposter	✗	3	0	1	1	13,821 (4.3%)	62,570 (8.4%)	2,871	3,614	4,737	9 (7)	112'42"
	locky	✗	7,076	0	7,074	1	9,546 (83.7%)	22,688 (51.5%)	89	7,303	715	15 (15)	22'59"
	samsam	✗	45	0	20	10	1,020 (29.7%)	1,108 (48.1%)	15	184	300	6 (6)	2'52"
	wannacry	✗	6	0	5	2	38 (78.9%)	531 (97.9%)	0	18	10	2 (2)	0'25"

Table 5: Back-and-forth explorations for the 30 malware families. In parenthesis, forward-only exploration results for comparison: fraction of addresses and transactions and total number of relations present in the forward-only graph.

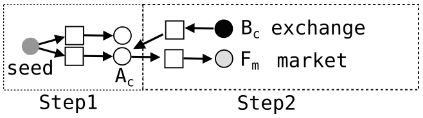

Figure 3: False positive example.

Surprisingly, forward-only exploration finds more relations (150) than back-and-forth exploration (142). The additional relations are FPs introduced by the exchange classifier failing to identify low-volume exchange addresses. Figure 3 illustrates this. Step 1 in the exploration finds address A_c, belonging to MI cluster c, which has no tag and is not identified as an exchange. If back-and-forth exploration is used, at step 2 B_c and F_m are introduced. Address F_m has an underground market tag, while address B_c is classified as an exchange. Now, our platform marks A_c also as an exchange because it belongs to the same cluster c as exchange address B_c, and removes paths starting from A_c introduced in step 2. Thus, the back-and-forth exploration graph corresponds to the step 1 box and contains a relation with exchange A_c. However, if forward-only exploration is used, then step 2 only discovers the market address F_m. The exchange address B_c is not discovered because it would require exploring backwards from A_c. Thus, the forward-only exploration includes the path from the seed to the market, which is a false relation introduced by the exchange address A_c. While this issue can affect both explorations, it affects more often forward-only exploration as the probability of finding addresses in the same exchange cluster is smaller.

The rightmost column captures the exploration runtime. It ranges from a few seconds up to 2 hours, with an average of 22 minutes per operation. Forward-only exploration is faster with an average runtime of 7 minutes due to the fewer transactions it analyzes. However, Sections 6.1 and 7 show forward-only exploration misses important relations that can only be identified by exploring backwards.

6.1 Evaluating Address Discovery

Evaluating the accuracy of the produced graphs is challenging as we lack ground truth on the malicious operations. However, for the four families that use the Bitcoin blockchain for C&C we are able to build operation oracles that can identify previously unknown C&C signaling addresses in the output graph, demonstrating the benefits of back-and-forth exploration.

The use of the Bitcoin blockchain for C&C follows a similar pattern for the four families. Samples have a hardcoded signaling address. Upon infection, the sample queries a external Bitcoin monitoring service (e.g., [10]) to obtain the most recent transactions involving the signaling address. Pony and Skidmap use the same signaling method that encodes C&C IP addresses in the value of two consecutive deposits into the signaling address [66]. Cerber encodes C&C domains in the value of a single transaction from the signaling address to a temporary address that then returns the funds [59]. And, Glupteba stores an encrypted C&C domain in the blockchain using a transaction from the signaling address with an OP_RETURN output [38]. In all cases, the signaling transactions create a distinct pattern for which we build an operation oracle. The oracle determines if a given address is a signaling address for the family. An exploration is performed from seeds obtained from previous reports and the operation oracle is applied to each address in the graph to identify previously unknown signaling addresses.

Cerber. The Cerber ransomware was the first malware to use the Bitcoin blockchain for C&C between July 2016 and late 2017 when the operators were arrested [70]. The Cerber oracle captures that a signaling round creates cyclic transactions where a signaling address sends x BTCs to a temporary address (only used for one signaling round) that then, within minutes, returns $x - fee$ to the signaling address, where *fee* is the Bitcoin transaction fee (i.e., 0.001

Address	In	Step	Found	FQDNs	Signaling Period
17gd1m	[59]	0	Seed	254	2016-07-26 ⟶ 2017-06-13
1HTDy9	[59]	1	MI	128	2017-02-13 ⟶ 2017-07-23
1ML94w	[59]	1	Forw	34	2017-06-11 ⟶ 2017-09-16
1CpTCV	[59]	1	MI	9	2017-08-31 ⟶ 2017-09-22
1GcnsL	✗	1	MI	17	2017-07-27 ⟶ 2017-09-22
14ru2h	✗	2	Back	4	2017-01-21 ⟶ 2017-02-13
1DiRvu	✗	2	Back	4	2017-03-12 ⟶ 2017-04-21

Table 6: Cerber signaling addresses. Exploring from 4 seeds, 3 previously unknown signaling addresses were discovered.

Address	In	Step	Found	IPs	Signaling Period
1BkeGq	[66]	0	Seed	237	2019-08-28 ⟶ 2021-01-13
1CeLgF	[66]	0	Seed	99	2019-08-27 ⟶ 2019-12-11
19hi8B	[66]	0	Seed	336	2019-12-10 ⟶ 2021-01-13
1N9ALZ	[3]	0	Seed	19	2019-10-03 ⟶ 2020-08-06
bc1qwl	✗	2	Forw	2	2019-07-24 ⟶ 2019-07-24
1NL8QT	✗	2	Forw	3	2020-03-03 ⟶ 2020-07-22
1GUghN	✗	2	Back	6	2020-03-03 ⟶ 2020-07-11
bc1qh9	✗	2	Back	24	2019-06-09 ⟶ 2019-08-26
bc1q0n	✗	2	Back	2	2019-12-09 ⟶ 2019-12-09
bc1qzs	✗	2	Back	15	2019-06-29 ⟶ 2019-07-21
35KxqL	✗	3	Back	2	2019-12-10 ⟶ 2019-12-10
3Dmcs6	✗	3	Back	1	2019-12-09 ⟶ 2019-12-09

Table 7: Pony signaling addresses. Exploring from 4 seeds, 8 previously unknown signaling addresses were discovered.

BTC). The first six characters of the temporary address correspond to a DNS domain on a specific TLD hardcoded in the sample. Given an address, it iterates over its withdrawal transactions. For each transaction, it performs five checks. First, it checks the transaction has a single input and a single output address. Then it verifies that the recipient is a temporary address, i.e., has a single deposit transaction and a single withdrawal transaction. Next, it validates that the recipient returns the received funds to the sender by checking that the withdrawal transaction has just one input and one output, that its output is the input address, and that it returns the value it received minus the transaction fee. The Cerber exploration starts from four signaling addresses mentioned in [59] and identifies three previously unknown signaling addresses. All seven addresses are detailed in Table 6. For each signaling address, the table shows whether it was known, the exploration step at which it was discovered, the method used to discover it, the number of domains it signaled, and the time period in which it signaled. The three signaling addresses discovered are not mentioned in [59], although they were active simultaneously with other seeds. One is identified by MI clustering, but identifying the other two required exploring backwards. This highlights how back-and-forth exploration is instrumental to identify new addresses.

Pony. The Pony downloader and information stealer started using the Bitcoin blockchain for signaling C&C servers in August 2019 [66]. Given an input address, the oracle tries to find pairs of consecutive transactions that deposit into the address, whose values encode a public (i.e., not private or reserved) IP address using a known encoding [30]. Each transaction in a valid pair encodes two bytes of the IP address. The oracle iterates over all transactions of the input address. Each deposit transaction should have at most three inputs and at most two outputs. The oracle temporarily stores the value of the first transaction and validates the transaction values if the second transaction happens within an hour of the first one. Both transaction values (in satoshis) are passed to the *decode* function described in [30]. If a public IP address is decoded, it is stored. The current pair is reset if a non-public IP address is found, any check fails, or if a withdrawal transaction is encountered since the two deposit transactions should be consecutive. If at least half of all deposit transactions received by the input address signal a public IP address, the input address is considered a signaling address. This last check handles high volume addresses where a pair of consecutive deposits may decode into a public IP address by chance. It also accounts for signaling addresses that were used for other purposes before they started signaling.

Address	In	Step	Found	FQDNs	Signaling Period
15y7ds	[55]	0	Seed	8	2019-06-19 ⟶ 2020-05-13
1CgPCp	[55]	0	Seed	6	2020-04-08 ⟶ 2021-10-19
1CUhaT	[40]	0	Seed	6	2021-10-13 ⟶ 2021-12-29
34Rqyw	✗	1	Forw	1	2020-01-23 ⟶ 2020-01-23
3NhC1b	✗	2	Forw	1	2020-01-24 ⟶ 2020-01-24

Table 8: Glupteba signaling addresses. Exploring from 3 seeds, 2 previously unknown addresses were discovered.

An exploration is performed starting from four seeds obtained from two sources [3, 66]. The operation oracle identifies 8 previously unknown signaling addresses. The 12 signaling addresses are detailed in Table 7. Discovering 6 of the 8 new addresses requires to explore at least one backwards step, which highlights the importance of back-and-forth exploration. Four of the new addresses are SegWit (starting prefix *bc*1). They seem to be used for testing before switching to the production non-SegWit addresses (starting prefix 1). For example, *bc*1*q*0*n* generates two IP addresses on December 9th, 2019. The next day, 19*hi*8*B* starts signaling IP addresses and the last IP signaled by *bc*1*q*0*n* is the first IP signaled by 19*hi*8*B*. The other SegWit addresses behave similarly. Using SegWit addresses for testing could be motivated by smaller transaction fees.

In addition to the new signaling addresses, backwards exploration also finds that funds used by two signaling addresses come from the MINE exchange [15], thus providing a previously unknown attribution point. This illustrates how backwards exploration can find important relations that forward transaction tracing cannot.

Encoding the C&C servers on deposit transactions has the downside that analysts can deposit funds into a signaling address to hijack the C&C communication. On August 14th, 2020, Taniguchi et al. [66] performed a takeover on Pony by signaling a server under their control to 1*Bke*4*Gq*. However, they did not sustain the takeover, so the attackers regained control two days later.

Skidmap. Skidmap is a crypto mining malware targeting Linux hosts [41]. On February 2021, Akamai researchers identified that Skidmap had started using Pony’s signaling as a backup C&C mechanism, and mentioned one signaling address [64]. We reuse Pony’s oracle for Skidmap. That seed is an online wallet. Thus, forward tracing from the seed cannot be performed. But, the exploration can proceed backwards from the seed. In this case the exploration does not find any new signaling address. However, in Section 7, we show how the backwards exploration identifies that the funds used in the signaling come from a previous Skidmap campaign and that operators cashed-out part of the funds in three exchanges that can serve as attribution points, showing how back-and-forth exploration reveals relations forward transaction tracing could not.

Glupteba. Glupteba is a botnet that has been operating since at least 2011 [36]. In September 2019, it was reported to have added Bitcoin as a mechanism to distribute backup C&C domains [38]. The

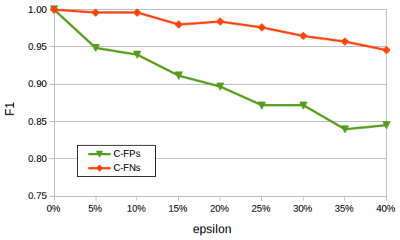

Figure 4: Relation detection impact when introducing exchange classifier errors. Introducing C-FPs produces R-FNs (green line). Introducing C-FNs produce R-FPs (red line).

C&C signaling leverages Bitcoin transactions with an OP_RETURN output, which stores in the blockchain an AES-GCM-256 encrypted backup C&C domain. The Glupteba oracle takes as input the address to validate and two decryption keys obtained from the malware binaries [55]. It first checks if the given address has a withdrawal transaction with an output with the field *data*, which indicates an OP_RETURN slot. If so, it checks that length of the data field is at least 56 hexadecimal characters (28 bytes). If so, it attempts to decrypt the data with the input keys, by using the first 12 bytes as the initialization vector, the last 16 bytes as the GCM tag, and the rest as the encrypted data. If the decryption succeeds, the decrypted plain text corresponds to the C&C domain.

An exploration is performed starting from three seeds obtained from prior reports [40, 55]. The operation oracle identifies 2 previously unknown signaling addresses in the graph. The 6 signaling addresses are detailed in Table 8. The two discovered addresses seem to have been used for testing. They signal the same domain in consecutive days and one day later that domain is signaled by address $15y7ds$. In December 2021, Google performed a take-down on the Glupteba botnet, although their note mentions that thanks to the Bitcoin C&C signaling the malware authors are likely to regain control of the botnet [40].

In summary, the results show back-and-forth exploration finds new C&C signaling addresses and important relations that forward transaction tracing could not. In particular, of the 13 new signaling addresses discovered, 8 are only discovered through backwards exploration, and the 4 attribution points found in Skidmap and Pony are also only discovered through backwards exploration.

6.2 Exchange Classifier Impact

This section evaluates the impact of the exchange classifier in the back-and-forth exploration. First, we measure how exchange classification errors impact the identification of end-to-end relations. Then, we perform an ablation study to measure the performance impact if the exchange classifier is not used. Both experiments are run on the 23 explorations with at least one seed that is not an online wallet.

Accuracy impact. A false positive by the exchange classifier (C-FP) might make the exploration discard valid paths and thus miss true relations, introducing end-to-end false negatives (R-FN). On the other hand, an exchange classifier false negative (C-FN) leads to larger graphs that may contain false relations introducing end-to-end false positives (R-FP). To measure the impact of exchange classification errors on the identification of end-to-end relations, we run the back-and-forth exploration multiple times on each family. Each exploration introduces errors to the exchange classifier results with an ϵ probability, and compares the relations found to the family's baseline in Table 5. An R-FN is the absence of a relation in the baseline. An R-FP is a new relation not present in the baseline.

	Runtime	Timeout	Addresses	TXES
With exchange classifier	499'37"	0	52,308	145,410
Without exchange classifier	2598'34"	7	243,019	787,397
Increase w/o exchange classifier	5.2x	7	4.6x	5.4x

Table 9: Total runtime and graph size, across the 23 families where at least one seed is not an online wallet, when running the exploration with and without the exchange classifier. To handle explosion when not using the classifier, 7 explorations were stopped when reaching 20K addresses.

For ease of interpretation, we perform two experiments: introducing only C-FPs and introducing only C-FNs. For example, when introducing C-FPs, if the classifier says an address is not an exchange, with ϵ probability the decision is flipped to mark it as an exchange. We increase ϵ between 5% and 40% in steps of 5%. Figure 4 shows, for each experiment, the F1 score of the relation detection as a function of ϵ. The results show that the introduction of C-FPs creates a more pronounced decrease in the relation detection results. Mis-classifying a non-exchange address as an exchange (C-FP) stops the exploration too early, which causes relations to be missed (R-FNs). On the other hand, C-FNs introduce additional addresses in the graph, but since many of those new addresses are untagged, the number of R-FPs introduced is lower in comparison, and thus the F1 score reduces more slowly. These results indicate that favoring a high precision of the exchange classifier is beneficial to the overall relation detection. This is already true for our classifier as Table 4 shows that its precision is higher than its recall.

Ablation study. This experiment runs the back-and-forth exploration on each family without using the exchange classifier. This is equivalent to the classifier introducing 100% C-FNs. The removal of the classifier causes some explorations to explode, i.e., they fail to finish within 12 hours. To be able to complete the experiment in reasonable time, we stop an exploration when its graph reaches 20k addresses. Table 9 summarizes the impact of removing the exchange classifier in terms of runtime and graph size. When the exchange classifier is not used, the graphs are on average 4.6 times larger and the runtime 5.2 times larger. In 7 of the 23 explorations, the 20K address limit is reached. Thus, the performance impact for those 7 explorations would be larger in reality. These results show how the exchange classifier is crucial to avoid explosion by limiting the runtime and graph size. The removal of the exchange classifier also causes the R-FPs to increase, as illustrated in the previous experiment, but the 20K address limit prevents us from precisely measuring the F1 decrease.

7 RELATIONSHIPS

Tagged addresses in a transaction graph may capture relations between the campaign under analysis and external services (e.g.,

Class	Operation	Relations Found	
C&C	cerber	1	malware:cerber:ransomnote
	glupteba	0	-
	pony	1	exchange:mineexchange
	skidmap	5	cryptojacking:sysupdate, exchange:bitfinex, exchange:coinbase, exchange:coincola*, exchange:huobi
Clipper	aggah	1	exchange:binance
	androidclipper	1	exchange:luno.com*
	azorult	1	exchange:tradeogre*
	bitcoingrabber	0	-
	casbaneiro	4	exchange:bleutrade.com, exchange:localbitcoins, exchange:exmo, payment:coinpayments
	clipbanker	1	exchange:coinbase*
	clipsa	6	exchange:binance, exchange:bitfinex, exchange:changenow, exchange:localbitcoins, mixer:wasabi, payment:coinpayments
	cliptomaner	1	exchange:tradeogre*
	cryptoshuffler	11	exchange:bittrex, exchange:btc-e.com, exchange:btctrade.com, exchange:cryptopay.me, exchange:cubits.com, exchange:exmo, exchange:poloniex, exchange:spectrocoin, mining:btcc-pool, payment:bitpay.com, payment:coinpayments
	masad	11	exchange:any-cash, exchange:binance, exchange:bitzlato, exchange:coinbase, exchange:cryptonator, exchange:localbitcoins, exchange:whitebit, exchange:z-exchange.ru, gambling:1xbit, payment:coinpayments, payment:payeer
	mekotio	6	clipper:mekotion40*, exchange:hitbtc, exchange:mercadobitcoin, miscabuse:mrpr0gr4mmer, mixer:wasabi, payment:coinpayments
	mekotion40	6	clipper:mekotio*, exchange:dsx, exchange:mercadobitcoin, gambling:foxbit, gambling:yolo-group, mixer:wasabi
	mispadu	1	exchange:mercadobitcoin*
	mrpr0gr4mmer	2	exchange:binance, payment:coinpayments
	n40	1	exchange:mercadobitcoin*
	phorpiex-tldr	3	exchange:cryptonator*, gambling:duckdice.io, miscabuse:saveyourself
	phorpiex-trik	5	exchange:bitcoin.de, exchange:localbitcoins, exchange:cryptonator*, gambling:rollin.io, payment:webmoney
	predatorthethief	8	clipper:masad, exchange:binance, exchange:bitzlato, exchange:btcbank, exchange:cryptonator, exchange:totalcoin, exchange:whitebit, payment:coinpayments
	protonbot	2	exchange:cryptonator*, exchange:hitbtc*
Ransomware	cryptotorlocker2015	11	exchange:bitcoin.de, exchange:cex.io, gambling:betcoin-dice, gambling:fortunejack.com, gambling:luckybit, gambling:satoshibones, gambling:satoshidice.com, mining:cointerra, mining:deepbit, mining:eobot, tormarket:silkroad2market
	cryptxxx	13	exchange:bitbay.net, exchange:bitcoin.de, exchange:bitfinex, exchange:bitflyer, exchange:btc-e.com, exchange:coinbase, exchange:coinmate.io, exchange:coins.co.th, exchange:huobi, exchange:localbitcoins, exchange:okcoin.com, exchange:poloniex, exchange:xapo
	dmalocker	8	exchange:btc-e.com, exchange:localbitcoins, exchange:okex, exchange:poloniex, gambling:cloudbet.com, malware:slave, payment:bitpay.com, payment:coinpayments
	globeimposter	9	exchange:bitstamp, exchange:bittrex, exchange:btc-e.com, exchange:cex.io, exchange:localbitcoins, payment:bitpay.com, payment:easywallet, payment:webmoney, tormarket:evolutionmarket
	locky	15	exchange:bitfinex, exchange:bitstamp, exchange:btc-e.com, exchange:coinbase, exchange:localbitcoins, exchange:matbea.com, exchange:poloniex, exchange:coins.co.th, exchange:exmo, exchange:xzzx.biz, mixer:bitcoinfog, payment:bitpay.com, payment:coinpayments, payment:webmoney, tormarket:nucleusmarket
	samsam	6	exchange:bitcoin.de, exchange:cubits.com, exchange:localbitcoins, exchange:poloniex, exchange:yobit.net, tormarket:alphabaymarket
	wannacry	2	exchange:changelly, exchange:shapeshift

Table 10: Summary of relationships found. Tags with an asterisk at the end were found by multi-input clustering on the seeds, the rest were found by the exploration.

exchanges, mixers, gambling, markets) and other malicious campaigns. Those relations are identified by the graph analysis module as paths that detail the specific chain of transactions capturing money flow from the campaign under analysis to an external entity, or vice-versa. Such paths constitute evidence that can be shared with third parties to allow independent validation of the relations.

Table 10 summarizes the relations found. In 93% (28/30) of the explorations at least one relation is identified, which we believe reflects that our tag database provides good coverage. Tags may have been assigned to addresses found by MI clustering on the seeds (9.2%) (marked with asterisk at the end of the tag) or during the exploration (90.8%). The large percentage difference highlights the need to perform transaction tracing (beyond MI clustering) to discover relations. The most common relations are with exchanges used by the operators for seeding C&C signaling and cashing out profits. We observe 44 exchanges, the most popular one being *localbitcoins* used by 9 families, followed by *coinbase* (5), *poloniex* (5), *cryptonator* (5), *btc-e* (5), and *binance* (5). Ransomware families use a larger variety of exchanges compared to clippers, e.g., the *cryptxxx* ransomware uses 13 exchanges. We also observe 5 payment processors, also used for cashing out: *coinpayments* (9), *bitpay* (4), *webmoney* (3), *easywallet* (1), and *payeer* (1). These services used for cashing out include some that are not amongst the most popular ones [68], which may be due to relaxed KYC measures. For example, *localbitcoins* is a peer-to-peer exchange that allows users to transact between them over-the-counter in a fairly anonymous manner. *Coinpayments* is a payment gateway that merchants can use to receive cryptocurrency with just an email and password. Both services did not establish KYC measures until mid-2019 [4, 6]. Furthermore, the owner of the *btc-e* exchange was arrested in July 2017 for money laundering with the indictment mentioning the lack of KYC measures [69].

Other common relations are gambling sites and mixers, likely used for money laundering. Six families are observed transacting with 11 gambling sites (e.g., *satoshidice*) with the *cryptotorlocker2015* ransomware using a record 5 gambling sites. Gambling sites are also possible attribution points since they may be subjected to legislation mandating KYC practices such as the Licence Conditions and Codes of Practice (LCCP) in the UK [27]. Two mixers appear across 4 families: *wasabi* (3) and *bitcoinfog* (1). Wasabi [18] is an open-source Bitcoin wallet implementing CoinJoin shuffling over Tor for anonymity. The operator of *bitcoinfog* was arrested on April 2021 on counts of unlicensed money transmission and money laundering [29]. Other relations found include three malware families using profits for buying products from four Tor hidden markets (*alphabaymarket*, *evolutionmarket*, *nucleusmarket*, *silkroad2market*) and two families transacting with four mining pools (*btcc-pool*, *cointerra*, *deepbit*, *eobot*). Finally, several relations between malicious campaigns are found, as detailed next.

Phorpiex-tldr and SaveYourself. The exploration from *phorpiex-tldr* reveals multiple connections with the *saveyourself* sextortion campaign. Figure 5 shows the paths capturing these relations, other paths are excluded for readability. There are six *saveyourself* tagged addresses in the graph, corresponding to addresses reported by Bitcoin Abuse users as appearing in sextortion emails with a common

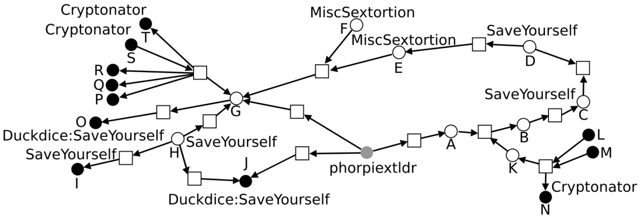

Figure 5: Phorpiex-tldr – SaveYourself relations.

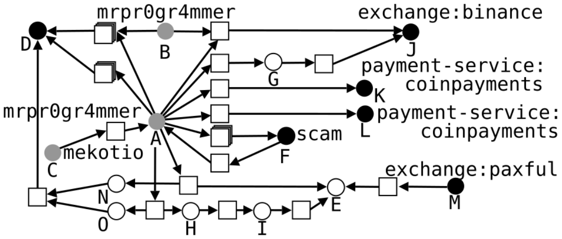

Figure 6: Mekotio-MrPr0gr4mmer relations.

SaveYourself pattern in the subject and content. The clipper's profits flow towards multiple *saveyourself* addresses through different paths, e.g., to tagged address C from the seed (in gray) through addresses A and B, to tagged address J from the seed, and through collector node G to tagged address O. Addresses O and J correspond to online wallets in the *duckdice.io* gambling service, managed by SaveYourself operators. Some connections are found through forward transaction tracing, but others such as the path going from the seed to address D through A, B, and C or the path going from the seed through G and H to reach node I require exploring backwards. This illustrates the benefits of back-and-forth exploration. A report [50] mentions that the Phorpiex malware sends sextortion emails. Checking the Bitcoin addresses in that report on Bitcoin Abuse confirms the link between Phorpiex and the SaveYourself campaign. Furthermore, the multiple connections between both campaigns and how the clipper profits flow into the sextortion addresses makes us conclude that the sextortion campaign is run by the malware operators, as opposed to the Phorpiex operators renting their botnet for sending third-party spam.

MrPr0gr4mmer clipper and scam. The MrPr0gr4mmer exploration, summarized in Figure 6, starts from two seeds (A, B) in the same MI cluster, each with multiple clipper reports in Bitcoin Abuse. Address F is reported in sextortion scam emails [7]. The sender of the scam emails is "Mr Robot". Manual searches uncover that seed A appears in the @MrPr0gr4mmer Telegram channel, where software to send spam through the telegram API [8] is advertised. The contact address starts again with the "Mr." prefix. Seed B appears as donation address for a user of the cracked.to forum and backwards tracing finds that collector E receives deposits from the *paxful* exchange. We conclude that the MrPr0gr4mmer operator could be attributed through the *paxful* exchange and the cracked.to forum and is involved in multiple malicious activities including selling spam tools, sending scam emails, and operating a clipper.

Mekotio, MekotioN40 and MrPr0gr4mmer. A seed from *mekotio* and a seed from *mekotion40* appear as inputs to the same transaction indicating that both families are indeed the same. While the relationship was already hinted by the names assigned in the two reports from where we obtained the seeds, those reports do not explicitly mention a connection. In addition, both graphs show cash outs to the *mercadobitcoin.br* exchange and transactions to the *wasabi* mixer in the non-shared components, further reinforcing that both campaigns have the same operators. Another relation is a payment from a *mekotio* seed to the seed of *mrpr0gr4mmer* used for selling the Telegram spam tool (see Figure 6). Thus, *mekotio* operators are using the clipper profits to pay for a spam tool possibly used as part of their malware distribution.

PredatorTheThief and Masad. The *predatorthethief* clipper seed makes a transaction to an address that a report identifies as belonging to the Masad Clipper and Stealer malware [44]. Masad is offered as a malware kit in forums. It has a free version and a full version that costs $85. The transaction on June 2020 is for 0.00910768 BTC, roughly $85.31 USD using the conversion at that time. We believe this relation captures how *predatorthethief* operators bought a license of *masad*, possibly to upgrade their own malware.

DMALocker and Slave. The *dmalocker* ransomware graph contains a transaction to an address tagged as belonging to the Slave banking trojan [24]. Slave receives 2.3 BTC from DMALocker, more than 1.3K USD at that time. Unfortunately, in this case we are not able to conclude what type of relation the transaction captures.

Skidmap and Sysupdate. The *skidmap* cryptominer backwards exploration (forward transaction tracing is not possible as the seed is an online wallet) identifies a *cryptojacking:sysupdate* address that sends funds to the seed six times. This address belongs to a mining pool used by a Linux cryptomining malware to receive the shares produced by the botnet's mining activity [5]. We believe this malware is an older *skidmap* version (prior to using Bitcoin for C&C) and that the relation shows that funds from Skidmap's cryptomining were used to seed the new Bitcoin C&C signaling. The Sysupdate address cashes out in three exchanges (*coinbase*, *huobi*, *bitmex*) that can be used as attribution points and are only found exploring backwards.

CryptoTorLocker2015 and Scam Campaigns. The MI cluster of the *cryptotorlocker2015* ransomware seeds contains 10 addresses beginning with short English words involved in multiple scam campaigns in 2014-2015. Four of those imitate satoshidice gambling addresses beginning with "1DiCE", another four (e.g., 1BonUS, 1sYSTEM) were used in scams that promise to double the BTCs sent [2], and two addresses beginning with "1bet" and "1shop" were used in campaigns claiming to have hacked the bitcointalk forum [1]. Those scams are likely run by the ransomware operators.

7.1 Limited Ground Truth Validation

Ground truth is not available for cybercrime operations and can only rarely be obtained, e.g., through leaks [45, 52]. Still, we tried hard to find some ground truth, even if limited, encountering two major challenges. First, prior analysis focuses on forward-only explorations and thus does not mention relations only reachable by exploring backwards. Second, analysis reports rarely exhaustively list relations, mentioning only those of interest to the report. We found only one report that partially addresses those challenges, which we use as a limited ground truth. It corresponds to the affidavit filed in April 2021 by the US Internal Revenue Service to justify the takedown of the *bitcoinfog* mixer [22]. The affidavit mentions *bitcoinfog* received deposits from 35 darknet markets, but only names five: *agora*, *alphabay*, *evolution*, *silkroad*, and *silkroad2*.

Our database has one address tagged as *bitcoinfog* belonging to a MI cluster with 244K addresses. Forward exploration from the mixer addresses does not work as the mixer's goal is to hamper forward transaction tracing. Thus, we perform a backwards-only exploration using as seed all addresses in the *bitcoinfog* MI cluster. The exploration finds a wealth of relations including all 15 darknet markets in our database. Those 15 markets include the five named in the affidavit and another 10: *abraxas*, *babylon*, *blackbank*, *bluesky*, *cannabisroad*, *doctord*, *middleearth*, *nucleus*, *pandora*, and *sheep*.

Since all 15 tagged darknet markets are found after one backwards step, we stop the exploration at that point. In addition to the darknet markets, the backwards exploration identifies relations with 130 services and 15 malicious operations not mentioned in the affidavit: 4 ransomware (*cryptolocker*, *locky*, *xlocker*, *xtplocker*), 3 scams (*mintpal*, *mcxnow*, *bitoomba*), 3 thefts (*binance-hack*, *bips-hack*, *flexcoin-hack*). 3 mixers (*bitlaunder*, *wasabi*, and an unknown mixer), a malware (*slave*), and a ponzi scheme (*nanoindustryinv*).

This experiment shows backwards exploration finds correct relations, although it may miss relations for which no tags are available. Unfortunately, we cannot evaluate FPs as the affidavit does not list all markets, and other services and malicious operations.

8 DISCUSSION

Backwards exploration. Our results show that back-and-forth exploration can discover important relations that forward transaction tracing cannot. One limitation is that backwards exploration is not suitable for every transaction. However, we show how it is possible to identify the problematic cases. For example, we provide a heuristic to identify dust attacks that may deposit small amounts to many addresses. We also provide an option to disable backwards exploration on seeds, which requires an easy decision by the analyst: use it when seeds may get deposits from other addresses belonging to the same campaign (e.g., C&C signaling) whereas avoid it when deposits come from victims (e.g., clippers, ransomware). Furthermore, even if a transaction was included that it should not, the transaction and its dependencies can later be pruned from the graph.

Exchange defenses. 90% of malware families analyzed had at least one relation with an exchange. This includes 10 malware families where addresses hardcoded in the malware correspond to online wallets in exchanges, These exchanges are either unaware about the abusive nature of those addresses, or they simply do not care. This creates an opportunity for intervention where public malware analysis reports could be automatically analyzed to extract tagged Bitcoin addresses (e.g., using indicator extraction approaches [48, 72]). The identified addresses could then be automatically explored, and those identified as online wallets or receiving cash-outs from the malware activities could be reported to the exchanges. This would prevent exchanges from claiming they were unaware of the abuse, and allow evaluating their reaction to abuse reports.

Address classification. Tagging Bitcoin addresses is fundamental to identify relations. A well-known tagging limitation is low coverage, as only a very small percentage of addresses is tagged. Furthermore, tagging is often delegated to commercial services (e.g., [13, 14]), which use proprietary approaches to build their tag databases and do not disclose how they handle challenges such as double-ownership and potentially conflicting tags. We believe open research on building large, accurate, tag databases is fundamental to allow tagging to sustain challenges likely to appear in court cases. Furthermore, we have shown how to build classifiers to identify certain types of addresses such as those belonging to exchanges and those used by malware for C&C signaling. A possible avenue of future work is to build classifiers for many other address types.

Evasion. An attacker could evade our approach by making its addresses look uninteresting, e.g., like exchanges. However, feature analysis shows that the high volume of transactions is a key feature of the exchange classifier, so impersonating an exchange would be expensive. Alternatively, an attacker could pollute our tag database so that some of its addresses are considered benign services. Evasion is also possible by using mixers since they prevent forward exploration. However, as shown, backwards exploration reveals their clients. Another evasion would pollute the graphs with unrelated transactions, e.g., by using sophisticated dusting techniques that our approach may fail to filter. This would make the graphs less readable, but would not prevent automated extraction of relations.

Other blockchains. Our approach generalizes to other cryptocurrencies following Bitcoin's model that are already supported by BlockSci: Bitcoin Cash, Dash, Litecoin, Namecoin, and ZCash. Supporting those would require extending our tag database with addresses for those blockchains and to retrain the exchange classifier.

9 CONCLUSION

Our study of the financial relationships of 30 malware families sheds light on cybercrime's financial inner workings. Our novel back-and-forth exploration identifies important relations missed by forward transaction tracing such as previously unknown C&C signaling addresses, a wealth of relations to external services and other cybercrime operations, and new attribution points. The results of this work can benefit the community in two main ways: (i) the open source tool and tag dataset we will provide may serve as basis for future works; and (ii) the detailed graphs and relations our approach produces could be used as evidence to warrant interventions.

ACKNOWLEDGMENTS

This work was funded by the Madrid regional government program S2018/TCS-4339 (BLOQUES-CM) and by the Spanish Governement through the SCUM Project (RTI2018-102043-B-I00) MCIN/AEI/10.13039/501100011033/ERDF and grants PRE2019-088472/MCIN/AEI/10.13039/501100011033 and IJC2020-043391-I/MCIN/AEI/10.13039/501100011033. Those projects are co-funded by European Union ESF, EIE, and NextGeneration funds. Partial support was also provided by ONR project HACRYPT (N00014-19-1-2292).

REFERENCES

[1] 2014. BTC-E and Bitcointalk.org Hacked? https://www.reddit.com/r/Bitcoin/comments/2mc8nc.

[2] 2014. Warning: New email scam people are apparently falling for - "secret doubling address". https://www.reddit.com/r/Bitcoin/comments/2fq3uc.

[3] 2019. Brief analysis of Redaman Banking Malware (v0.6.0.2) Sample. http://www.peppermalware.com/2019/11/brief-analysis-of-redaman-banking.html.

[4] 2019. Coinpayments - Service Restriction Notice. https://bitcointalk.org/index.php?topic=5156686.

[5] 2019. Linux mining virus-sysupdate. https://programmerclick.com/article/47131149819/.

[6] 2019. LocalBitcoins Statement on the Coming AML regulations and Compliance. https://localbitcoins.com/blog/aml-regulations-compliance.

[7] 2020. Bitcoin Porn Scam and Sextortion. https://www.onlinethreatalerts.com/article/2020/4/9/bitcoin-porn-scam-and-sextortion/.

[8] 2020. pr0fessionalcrackers telegram channel. https://ru.tgchannels.org/channel/MrPr0gr4mmer.

[9] 2022. Bitcoin Abuse. https://www.bitcoinabuse.com.

[10] 2022. Bitcoin Explorer. https://www.blockchain.com/explorer.

[11] 2022. Bitcoin OTC. https://bitcoin-otc.com.

[12] 2022. Bitcoin Talk Forum. https://bitcointalk.org.

[13] 2022. Chainalysis. https://www.chainalysis.com/.

[14] 2022. Elliptic. https://www.elliptic.co/.

[15] 2022. MINE.exchange. https://mine.exchange.

[16] 2022. Satoshi DICE. https://satoshidice.com/.

[17] 2022. Wallet Explorer. https://www.walletexplorer.com/info.

[18] 2022. Wasabi Wallet. https://wasabiwallet.io/.

[19] 2022. Watch your back repository. https://github.com/cybersec-code/watchyourback.

[20] Elli Androulaki, Ghassan O. Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. 2013. Evaluating User Privacy in Bitcoin. In *Financial Cryptography and Data Security*.

[21] M. Bartoletti, B. Pes, and S. Serusi. 2018. Data Mining for Detecting Bitcoin Ponzi Schemes. In *Crypto Valley Conference on Blockchain Technology*.

[22] Devon Beckett. 2021. Bitcoin Fog Affidavit by Internal Revenue Service. https://storage.courtlistener.com/recap/gov.uscourts.dcd.230456/gov.uscourts.dcd.230456.1.1$_1$.pdf.

[23] Juan Caballero, Gibran Gomez, Srdjan Matic, Gustavo Sánchez, Silvia Sebastián, and Arturo Villacañas. 2022. GoodFATR: A Platform for Automated Threat Report Collection and IOC Extraction. *CoRR* abs/2208.00042 (July 2022). https://doi.org/10.48550/arXiv.2208.00042

[24] CERT Polska. 2015. Case Study of Malicious Actors: Going Postal. https://maldroid.github.io/docs/The$_\text{P}$ostal$_\text{G}$roup.pdf.

[25] Nicolas Christin. 2013. Traveling the Silk Road: A measurement analysis of a large anonymous online marketplace. In *The World Wide Web Conference*.

[26] Richard Clayton, Tyler Moore, and Nicolas Christin. 2015. Concentrating Correctly on Cybercrime Concentration. In *Workshop on Economics of the Information Society*.

[27] UK Gambling Commission. 2020. LCCP Condition 17.1.1 - Customer identity verification. https://www.gamblingcommission.gov.uk/licensees-and-businesses/lccp/condition/17-1-1-customer-identity-verification.

[28] Mauro Conti, Ankit Gangwal, and Sushmita Ruj. 2018. On the Economic Significance of Ransomware Campaigns: A Bitcoin Transactions Perspective. *Computers & Security* 79 (2018), 162–189.

[29] Nikhilesh De. 2021. US Officials Arrest Alleged Operator of $336M Bitcoin Mixing Service. https://www.coindesk.com/us-officials-arrest-alleged-operator-of-336m-bitcoin-mixing-service.

[30] Kobi Eisenkraft and Arie Olshtein. 2019. Pony's C&C servers hidden inside the Bitcoin blockchain. https://research.checkpoint.com/2019/ponys-cc-servers-hidden-inside-the-bitcoin-blockchain/.

[31] Europol. 2019. Multi-million euro cryptocurrency laundering service Bestmixer.io taken down. https://www.europol.europa.eu/newsroom/news/multi-million-euro-cryptocurrency-laundering-service-bestmixerio-taken-down.

[32] Martin Gill and Geoff Taylor. 2004. Preventing Money Laundering or Obstructing Business? Financial Companies' Perspectives on 'Know Your Customer' Procedures. *British Journal of Criminology* 44, 4 (2004), 582–594.

[33] Steven Goldfeder, Harry A. Kalodner, Dillon Reisman, and Arvind Narayanan. 2018. When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies. *PoPETs* 2018 (2018), 179–199.

[34] Gibran Gomez, Pedro Moreno-Sanchez, and Juan Caballero. 2022. Watch Your Back: Identifying Cybercrime Financial Relationships in Bitcoin through Back-and-Forth Exploration. *CoRR* abs/2206.00375 (June 2022). https://doi.org/10.48550/ARXIV.2206.00375

[35] Mikkel Alexander Harlev, Haohua Sun Yin, Klaus Christian Langenheldt, Raghava Rao Mukkamala, and Ravi Vatrapu. 2018. Breaking Bad: De-Anonymising Entity Types on the Bitcoin Blockchain Using Supervised Machine Learning. In *Hawaii International Conference on System Sciences*.

[36] David Harley and Aleksandr Matrosov. 2011. TDL4 and Glupteba: Piggyback PiggyBugs. https://www.welivesecurity.com/2011/03/02/tdl4-and-glubteba-piggyback-piggybugs/h.

[37] Bernhard Haslhofer, Roman Karl, and Erwin Filtz. 2016. O Bitcoin Where Art Thou? Insight into Large-Scale Transaction Graphs. In *SEMANTiCS Conference*.

[38] Jaromir Horejsi and Joseph C Chen Salgado. 2019. Glupteba Hits Routers and Updates C&C Servers. https://www.trendmicro.com/en$_\text{u}$s/research/19/i/glupteba-campaign-hits-network-routers-and-updates-cc-servers-with-data-from-bitcoin-transactions.html.

[39] D. Y. Huang, M. M. Aliapoulios, V. G. Li, L. Invernizzi, E. Bursztein, K. McRoberts, J. Levin, K. Levchenko, A. C. Snoeren, and D. McCoy. 2018. Tracking Ransomware End-to-end. In *IEEE Symposium on Security and Privacy*.

[40] Shane Huntley and Luca Nagy. 2021. Disrupting the Glupteba operation. https://blog.google/threat-analysis-group/disrupting-glupteba-operation/.

[41] Augusto Remillano II, Jakub Urbanec, and Wilbert Luy Saias. 2019. Skidmap Malware Uses Rootkit to Hide Mining Payload. https://www.trendmicro.com/en$_\text{u}$s/research/19/i/skidmap-linux-malware-uses-rootkit-capabilities-to-hide-cryptocurrency-mining-payload.html.

[42] Marc Jourdan, Sebastien Blandin, Laura Wynter, and Pralhad Deshpande. 2018. Characterizing Entities in the Bitcoin Blockchain. In *IEEE International Conference on Data Mining Workshops*.

[43] Harry Kalodner, Malte Möser, Kevin Lee, Steven Goldfeder, Martin Plattner, Alishah Chator, and Arvind Narayanan. 2020. BlockSci: Design and Applications of a Blockchain Analysis Platform. In *USENIX Security Symposium*.

[44] Paul Kimayong. 2019. Masad Stealer: Exfiltrating using Telegram. https://blogs.juniper.net/en-us/threat-research/masad-stealer-exfiltrating-using-telegram.

[45] Brian Krebs. 2022. Conti Ransomware Group Diaries, Part I: Evasion. https://krebsonsecurity.com/2022/03/conti-ransomware-group-diaries-part-i-evasion/.

[46] Seunghyeon Lee, Changhoon Yoon, Heedo Kang, Yeonkeun Kim, Yongdae Kim, Dongsu Han, Sooel Son, and Seungwon Shin. 2019. Cybercriminal Minds: An Investigative Study of Cryptocurrency Abuses in the Dark Web. In *Network and Distributed Systems Security Symposium*.

[47] K. Liao, Z. Zhao, A. Doupe, and G. Ahn. 2016. Behind Closed Doors: Measurement and Analysis of CryptoLocker Ransoms in Bitcoin. In *APWG Symposium on Electronic Crime Research*.

[48] Xiaojing Liao, Kan Yuan, XiaoFeng Wang, Zhou Li, Luyi Xing, and Raheem Beyah. 2016. Acing the IOC Game: Toward Automatic Discovery and Analysis of Open-Source Cyber Threat Intelligence. In *ACM SIGSAC Conference on Computer and Communications Security*.

[49] Y. Lin, P. Wu, C. Hsu, I. Tu, and S. Liao. 2019. An Evaluation of Bitcoin Address Classification based on Transaction History Summarization. In *IEEE International Conference on Blockchain and Cryptocurrency*.

[50] Gil Mansharov and Alexey Bukhteyev. 2019. In the footsteps of a sextortion campaign. https://research.checkpoint.com/2019/in-the-footsteps-of-a-sextortion-campaign/.

[51] Gregory Maxwell. 2013. CoinJoin: Bitcoin privacy for the real world. https://bitcointalk.org/index.php?topic=279249.0.

[52] Damon McCoy, Andreas Pitsillidis, Jordan Grant, Nicholas Weaver, Christian Kreibich, Brian Krebs, Geoffrey Voelker, Stefan Savage, and Kirill Levchenko. 2012. PharmaLeaks: Understanding the Business of Online Pharmaceutical Affiliate Programs. In *USENIX Security Symposium*.

[53] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage. 2013. A Fistful of Bitcoins: Characterizing Payments among Men with No Names. In *Internet Measurement Conference*.

[54] M. Möser, R. Böhme, and D. Breuker. 2013. An Inquiry into Money Laundering Tools in the Bitcoin Ecosystem. In *APWG eCrime Researchers Summit*.

[55] Luca Nagy. 2020. Glupteba: Hidden Malware Delivery in Plain Sight. https://news.sophos.com/wp-content/uploads/2020/06/glupteba$_\text{f}$inal.pdf.

[56] Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. (2008). https://bitcoin.org/bitcoin.pdf.

[57] Masarah Paquet-Clouston, Bernhard Haslhofer, and Benoit Dupont. 2019. Ransomware Payments in the Bitcoin Ecosystem. *Journal of Cybersecurity* 5, 1 (2019).

[58] Masarah Paquet-Clouston, Matteo Romiti, Bernhard Haslhofer, and Thomas Charvat. 2019. Spams Meet Cryptocurrencies: Sextortion in the Bitcoin Ecosystem. In *ACM Conference on Advances in Financial Technologies*.

[59] S. Pletinckx, C. Trap, and C. Doerr. 2018. Malware Coordination using the Blockchain: An Analysis of the Cerber Ransomware. In *IEEE Conference on Communications and Network Security*.

[60] Rebecca S. Portnoff, Danny Yuxing Huang, Periwinkle Doerfler, Sadia Afroz, and Damon McCoy. 2017. Backpage and Bitcoin: Uncovering Human Traffickers. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[61] Dorit Ron and Adi Shamir. 2013. Quantitative Analysis of the Full Bitcoin Transaction Graph. In *Financial Cryptography and Data Security*.

[62] Dorit Ron and Adi Shamir. 2014. How Did Dread Pirate Roberts Acquire and Protect his Bitcoin Wealth?. In *Financial Cryptography and Data Security*.

[63] Jan Rubín. 2019. Clipsa - Multipurpose password stealer. https://decoded.avast.io/janrubin/clipsa-multipurpose-password-stealer/.

[64] Evyatar Saias. 2021. Bitcoins, blockchains, and botnets. https://blogs.akamai.com/sitr/2021/02/bitcoins-blockchains-and-botnets.html.

[65] Michele Spagnuolo, Federico Maggi, and Stefano Zanero. 2014. BitIodine: Extracting Intelligence from the Bitcoin Network. In *Financial Cryptography and Data Security*.

[66] Tsuyoshi Taniguchi, Harm Griffioen, and Christian Doerr. 2021. Analysis and Takeover of the Bitcoin-Coordinated Pony Malware. In *ACM ASIA Conference on Computer and Communications Security*.

[67] Ege Tekiner, Abbas Acar, A. Selcuk Uluagac, Engin Kirda, and Ali Aydin Selçuk. 2021. SoK: Cryptojacking Malware. In *IEEE European Symposium on Security and Privacy*.

[68] Taylor Tepper and John Schmidt. 2021. Best Crypto Exchanges For 2021. https://www.forbes.com/advisor/investing/best-crypto-exchanges/.

[69] US Department of Justice. 2017. Russian National And Bitcoin Exchange Charged In 21-Count Indictment For Operating Alleged International Money Laundering Scheme And Allegedly Laundering Funds From Hack Of Mt. Gox. https://www.justice.gov/usao-ndca/pr/russian-national-and-bitcoin-exchange-charged-21-count-indictment-operating-alleged.

[70] Shaun Waterman. 2017. European police take down criminals behind two big ransomware strains. https://www.cyberscoop.com/ctb-locker-cerber-ransomware-arrests-europol-mcafee/.

[71] Haohua Sun Yin and Ravi Vatrapu. 2017. A First Estimation of the Proportion of Cybercriminal Entities in the Bitcoin Ecosystem using Supervised Machine Learning. In *IEEE International Conference on Big Data*.

[72] Ziyun Zhu and Tudor Dumitras. 2018. ChainSmith: Automatically Learning the Semantics of Malicious Campaigns by Mining Threat Intelligence Reports. In *IEEE European Symposium on Security and Privacy*.