Toward Understanding Asset Flows in Crypto Money Laundering Through the Lenses of Ethereum Heists

Jiajing Wu, *Senior Member, IEEE*, Dan Lin, *Graduate Student Member, IEEE*, Qishuang Fu, Shuo Yang, *Graduate Student Member, IEEE*, Ting Chen, *Member, IEEE*, Zibin Zheng, *Fellow, IEEE*, and Bowen Song

***Abstract*— With the overall momentum of the blockchain industry, financial crimes related to blockchain crypto-assets are becoming increasingly prevalent. After committing a crime, the main goal of cybercriminals is to obfuscate the source of the illicit funds in order to convert them into cash and get away with it. Many studies have analyzed money laundering (ML) in the field of the traditional financial sector. However, in terms of the emerging blockchain crypto-asset ecosystem, there is currently only one public anti-money laundering (AML) dataset for Bitcoin – the Elliptic dataset, whose binary labels (licit vs. illicit transactions) cannot cover the ML behaviors in the evergrowing crypto-asset market. To fill this gap, in this paper, we propose a framework named XBlockFlow which identifies ML addresses starting from Ethereum heist incidents and obtains the first detailed Ethereum ML dataset named *EthereumHeist*, and then conducts a comprehensive feature and evolution analysis on the *EthereumHeist* dataset according to the three main phases of ML. We first search for the source cybercriminal accounts including exchange hackers, DeFi exploiters, and scammers. Then, employing the idea of taint analysis, we track the diverse downstream transactions and addresses layer by layer. At the end of tracking, we identify and categorize service providers, and go a step further to investigate advanced ML methods that do not exist in the Bitcoin scenario, e.g. token swap and counterfeit token creation. Based on the ML identification results, we obtain many interesting findings about crypto-asset money laundering, observing the escalating money laundering methods such as creating counterfeit tokens and masquerading as speculators.**

***Index Terms*— Blockchain, cryptocurrency, anti-money laundering.**

Manuscript received 19 July 2023; revised 8 November 2023; accepted 12 December 2023. Date of publication 22 December 2023; date of current version 29 December 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62332004, Grant 62372485, Grant 61973325, Grant 62032025, and Grant 61973325; and in part by the Natural Science Foundations of Guangdong Province under Grant 2023A1515011314. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Edgar Weippl. *(Corresponding authors: Dan Lin; Ting Chen.)*

Jiajing Wu, Dan Lin, Shuo Yang, and Zibin Zheng are with the School of Software Engineering, Sun Yat-sen University, Zhuhai 519082, China (e-mail: lind8@mail2.sysu.edu.cn).

Qishuang Fu is with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China.

Ting Chen is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: brokendragon@uestc.edu.cn).

Bowen Song is with Ant Group, Hangzhou 310013, China.

Digital Object Identifier 10.1109/TIFS.2023.3346276

I. Introduction

THE past decade has witnessed the rapid growth of blockchain and the blockchain-based cryptocurrency ecosystem. The market capitalization of cryptocurrencies has reached a staggering scale, with Bitcoin reaching a market capitalization of $385 Billion [1]. With lots of upgrades and more advanced features, blockchain technology has grown from Blockchain 1.0 to Blockchain 3.0, whose applications range from virtual currencies, to smart contracts, to decentralized society [2], [3]. The current blockchain financial ecosystem is built on three essential fundamentals: an underlying blockchain that stores transaction records and ensures the decentralized nature, smart contracts that represent the logic of the application, and crypto-assets[1] that can represent anything of value.

However, since blockchain transactions do not require user identification information, blockchain and crypto-asset ecosystem have become a hotbed of various cybercrimes and illegal financial activities [5]. According to a blockchain security firm named Chainalysis [6], the amount of crypto-assets stolen rises from $0.5 billion in 2020 to $3.3 billion in 2021 and $3.8 billion by 2022. After stealing crypto-assets, cybercriminals conceal and disguise them through different channels to make them appear legitimate and then withdraw them, a process known as money laundering (ML). So it is said that ML is the subsequent part of all other forms of crypto-based crimes [7], [8]. Therefore, with the frequent occurrence of blockchain security incidents, anti-money laundering (AML) for crypto-assets is in a crucial position to be the last line of defense to stop hackers from successfully cashing out and also to deter hackers from committing crypto crimes at the same time.

Anti-money laundering is not a new issue, and a wealth of research on the AML issue in traditional financial scenarios has been proposed [9], [10], [11], [12], [13]. In the field of cryptocurrency, the Elliptic dataset [4] is the first open-source bitcoin AML dataset that labels illicit/licit bitcoin transactions roughly, as displayed in Fig. 1. Many researchers have followed up with a series of crypto AML studies based

[1]Crypto-assets ("crypto" for short) refer to a broad category of digital assets, including cryptocurrencies, tokens, etc. Crypto-assets and cryptocurrencies are used interchangeably in this paper.

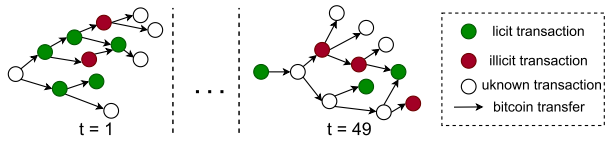


Fig. 1. Sample visualization of data from the Elliptic Bitcoin AML dataset [4]. The Elliptic dataset contains 49 time steps, from $t = 1$ to $t = 49$. Each vertex denotes a bitcoin transaction and each edge denotes the flow of bitcoin currency (i.e., BTC) going from one transaction to the next one. Transactions are deemed licit (versus illicit) if the input addresses of the transaction belong to a licit (versus illicit) category.

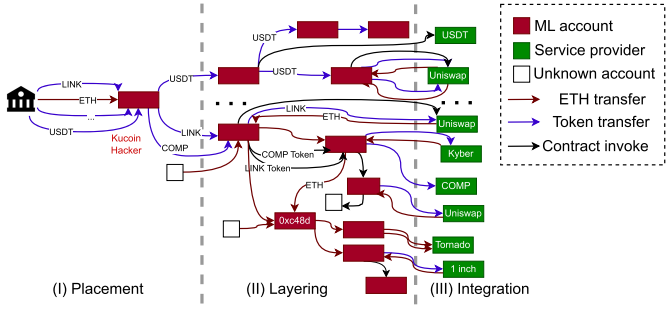


Fig. 2. Sample visualization of the money laundering process after Kucoin exchange hack on Ethereum which contains three phases: placement, layering, and integration. The ellipsis in a box means more than one account. Each vertex denotes an Ethereum account and each edge denotes the transfer of crypto-asset (i.e., ETH and tokens) with amounts and a timestamp. ETH transfers include external and internal transactions.

on this dataset [14], [15], [16]. But for one, this dataset only has binary labels for ML with no other business details, and for two, this dataset focuses on Bitcoin only, and does not reflect the diversity of stolen assets and the decentralization of service providers.

Fig. 2 shows the sample downstream transactions from an address labeled "Kucoin Hack", a real case that took place on Ethereum in 2020 [17]. Downstream ML transactions of this case can be summarized into the following three phases: ***(i)*** gather the several crypto-asset from Kucoin exchange and *place* to their own accounts; ***(ii)*** maximize the dispersion of illegally obtained funds through complex multiple *layered* transactions, including ETH transfers and token transfers; ***(iii)*** *aggregate* and withdraw the stolen funds in batches through service providers accounts. To the best of our knowledge, there is neither a publicly available dataset on money laundering in Ethereum, nor a framework for systematically collecting and analyzing the escalating money laundering activities.

However, AML approaches on traditional financial scenarios or Bitcoin cannot be directly applied to other blockchain platforms like Ethereum, due to three levels of challenges:

(i) The underlying blockchain. Compared with traditional finance, blockchain is decentralized, borderless, and pseudonymous, without limiting the number of accounts each user can create. This allows cybercriminals to conduct lots of frequent transactions between accounts under their control, leading to difficult identification of account entities and pseudonymous transfers. **(ii) Smart contracts.** Based on blockchain, smart contracts enable various types of crypto-assets which can be exchanged in the trading platforms. At the same time, Turing-complete smart contracts can represent and execute

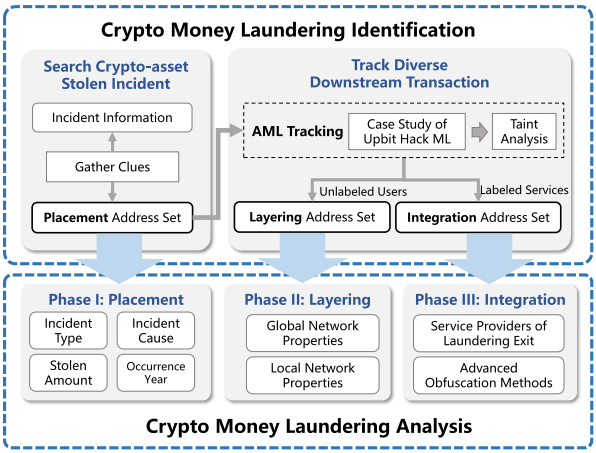


Fig. 3. The proposed XBlockFlow framework consists of two parts: identification and analysis.

more complex application logic and functions, leading to more complex transaction patterns. **(iii) Decentralized finance (DeFi)** [18]. On the one hand, immature DeFi applications gather a large number of assets, attracting the attention of criminals and becoming the hardest hit by asset theft; on the other hand, DeFi services lacking AML compliance bring ever-changing means of exchanging coins while also fueling crypto criminals to launder dirty money.

In this paper, we aim to explore the crypto-asset ML behaviors by taking Ethereum, the representative blockchain platform [1], as an entry point. Specifically, we first propose an abstract model to describe the crypto-asset ML and explain the relationship between AML and forensics in the field of blockchain (Section II). Then, following the crypto ML process, we build a framework named "XBlockFlow" to explore blockchain crypto-asset flow until destination service providers for ML identification and analysis[2], in order to carve out the full picture of the security incidents and complete the chain of evidence for the transfer of stolen assets. As shown in Fig. 3, our framework includes searching crypto-asset stolen incidents and placement addresses in Phase I, tracking diverse downstream transactions based on taint analysis in Phase II, and categorizing terminal service providers in Phase III (Section III). In the analysis part, we first show the selected representative incidents, including tagged accounts of hackers, exploiters, and scammers, and their basic information (Section IV-B). Then, comprehensive graph analysis is used to dig and understand the ML addresses of typical ML mechanisms (Section IV-C). Furthermore, we reveal the novel ML techniques based on the third-party service provider involved (Section IV-D). **The main contributions are as follows.**

- **Methodology.** To our best knowledge, we present the *first* anti-money laundering framework for Ethereum called XBlockFlow, which extracts money laundering addresses through the lenses of Ethereum heists from 2016–2022 and understands Ether money laundering through a

[2]The scope of identification in our proposed framework extends from the source addresses of stolen funds to various service providers. The ML activities after the destination service providers (such as cross-chain bridges or coin mixers will be discussed in our future work (Section V-A).

detailed staged analysis. The XBlockFlow framework is available online [19] and can also be reused for future study.

- **Dataset.** Based on the XBlockFlow framework, we collect a real-world dataset (containing over 160,000 addresses) that can be used for subsequent crypto AML studies, containing rich information: the time and amount of transactions, the label of the service provider address, the layer in which the ML address, etc. The dataset is also available online [19].
- **Insights.** We obtain many interesting findings of crypto ML by adopting feature analysis, graph analysis, and other methods. These findings help us gain new knowledge about crypto ML behaviors. Particularly, we find that it is common for exploiters to obfuscate stolen funds by swapping tokens through DeFi platforms, and hackers even launder money by creating counterfeit tokens for higher anonymity.

II. Preliminary

A. Background Information

1) Bitcoin and UTXO Model: The initial capitalization "Bitcoin" refers to the Bitcoin technology and Bitcoin platform, and the initial lowercase "bitcoin" refers to the currency itself [20]. An unspent transaction output (UTXO) model [21] is an account model used in the Bitcoin platform, which records account balances in the form of unspent transaction outputs (UTXOs).

2) Ethereum and Account Model: Ethereum [22] is a platform that enables the creation of custom financial products on the web. The account model [22] on Ethereum consists of externally owned accounts (EOAs) and Contract Accounts (CAs). An external transaction [23] is initiated by an EOA. An Internal transaction [23] is initiated by a CA, which results in the transfer of assets or the invocation of other contracts within external transactions.

3) Tokens: A token [24] is a digital asset implemented in a smart contract and is the medium for the storage and exchange of value in the blockchain. ERC20, ERC721, and ERC1155 [25] are currently widely used token contract standards. An ERC20 token transaction [26] is the circulation of ERC20 tokens. Common token contracts implement token management operations, such as minting, burning, and transfer.

4) Virtual Asset Service Provider (VASP): A virtual asset service provider (VASP) [27] provides services to convert cash between virtual assets and fiat currencies. A Centralized Exchange (CEX) [28] is a type of cryptocurrency exchange that is operated by a central authority or organization, while a Decentralized Exchange (DEX) [28] operates on a decentralized network and allows users to trade peer-to-peer or peer-to-pool without the need for a central authority or organization.

5) Normal Address: A normal address is one that is not reported to be involved in illegal activity.

6) Dirty Money: Dirty money [10] refers to funds that are obtained through illegal or illicit activities and are often used to conceal the source of the funds or to finance further criminal activities.

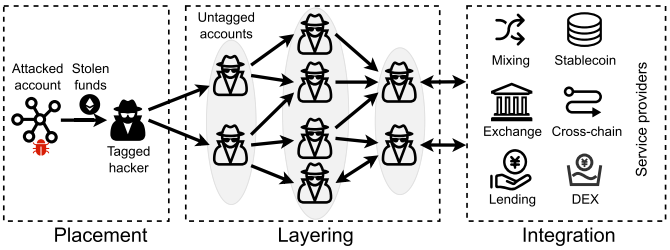


Fig. 4. The abstract model of the crypto money laundering process. An ellipse with a gray background represents a laundering layer.

B. Abstraction Model for Money Laundering

We first propose an abstract model of the crypto ML process. Formally, the money laundering process of a heist can be defined as a four-tuple:

$$(\mathcal{P}, \mathcal{L}, \mathcal{I}, \mathcal{T}),$$

where $\mathcal{P}$, $\mathcal{L}$ and $\mathcal{I}$ represent the address sets of *placement*, *layering*, and *integration*, respectively (corresponding to the three phases of money laundering mentioned in Section I). $\mathcal{T}$ is a transaction set that represents the involved transactions during the money laundering process, including external, internal, and ERC20 token transactions. Fig. 4 shows a toy example of crypto money laundering on Ethereum.

- **Placement phase:** the hacker performs an attack to steal assets and place them in $\mathcal{P}$, i.e., placement address set. The addresses in $\mathcal{P}$ are the source of the stolen funds.
- **Layering phase:** After taking $\mathcal{P}$, the hacker initiates multiple transactions of ETH or ERC20 tokens, passing the money in $\mathcal{P}$ layer by layer into the untagged layering address set $\mathcal{L}$ in the layering phase, cycling back and forth, obfuscating the source. Laundering layers refer to the collections of suspicious accounts laundered from the placement set circle after circle. The k-th laundering layer refers to the set of suspicious addresses at the same level k. The tracking level or depth of the layering phase is defined as the number of transaction hops from the original placement account. The number of laundering layers refers to the maximum tracking level or depth.
- **Integration phase:** Finally, the stolen funds are aggregated to integration address set $\mathcal{I}$ for cash out. The addresses in $\mathcal{I}$ are usually service providers such as exchanges, DeFi platforms, etc.

C. XBlockFlow for Crypto-Asset Anti-Money Laundering

Following the four-tuple abstract model of the crypto money laundering process, in this work, we propose a AML framework called XBlockFlow to detect and analyze ML behaviors on Ethereum, which mainly contains two parts, as shown in Fig. 3.

The first part of XBlockFlow is crypto ML identification, which refers to finding the ML addresses associated with the crypto stolen incidents according to the evidence of transactions. In this part, we (i) search crypto-asset stolen incidents via reports and websites, to obtain the placement address set, and (ii) track diverse downstream transactions via the proposed algorithm to demystify the layering address set and

integration address set. We formulate the identification task of the XBlockFlow framework described as follows: *Given the incident reported in the real world with placement address set $\mathcal{P}$ (Phase I), the goal is to collect the downstream transactions with the taint analysis (Phase II) and the aggregation address set including service providers (Phase III). The output of this part is the layering address set $\mathcal{L}$, the integration address set $\mathcal{I}$, and the transaction set $\mathcal{T}$.*

The second part of XBlockFlow is crypto ML analysis, which aims to investigate and characterize the identified crypto ML addresses according to the three main phases of ML. Specifically, (i) with *placement* address set $\mathcal{P}$ (Phase I), we describe and statistic the selected representative incidents and provide the collected information of incidents; (ii) with *layering* address set $\mathcal{L}$ (Phase II) and the transaction set $\mathcal{T}$, we model the layering transactions collection as a graph, and further measure both global and local network properties to understand the crypto ML behaviors and patterns; (iii) with *integration* address set $\mathcal{I}$, we go deeper to understand the evolution of service providers in the blockchain ecosystem to observe the evolution of crypto ML techniques.

III. Identification Methodology

In this section, we describe our identification methodology of XBlockFlow, including (i) searching crypto-asset stole incidents reported in the real world to identify their corresponding placement address set; (ii) tracking diverse downstream transactions to identify their corresponding layering address set and integration address set via a proposed algorithm guided by a real-world case study.

A. Search Crypto-Asset Stolen Incidents

Ethereum has witnessed many security incidents of crypto-asset theft each year since its creation. As of October 27, 2022, the "Label Word Cloud" service on Etherscan has flagged 115 addresses as "Heist".[3] This list of "Heist" addresses, which also contains the name tag corresponding to each address, indicates the address of the criminal who committed the attack or fraud that successfully stole the money for the case. Those 115 addresses are related to different types of stolen assets, such as exchange hacks, scam projects, DeFi exploits, etc. Thus, we model each address in Etherscan "Heist" list as the corresponding placement set $\mathcal{P}$ for each incident.

For better understanding, here we show a real example in Fig. 5, which displays one of the addresses in the Etherscan "Heist" address list, 0xeb31973e0febf3e3d705823-4a5ebbae1ab4b8c23, corresponding to the name tag "Kucoin Hacker", which implies that the address belongs to the hacker of the Kucoin exchange attack. Apart from Etherscan, these source heist addresses can be collected from official statements of victimized projects or media reports. An official statement about the Kucoin hack was released on the news channel of Kucoin's official website[4] by Kucoin Global CEO.

[3]https://etherscan.io/accounts/label/heist

[4]https://www.kucoin.com/news/en-kucoin-ceo-livestream-recap-latest-updates-about-security-incident

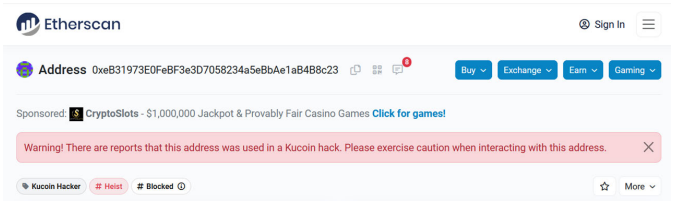


Fig. 5. An Etherscan screenshot of address tagged as "Kucoin Hacker". This address is alerted to the fact that it has been involved in a Kucoin exchange attack and is also marked with a "#Heist" tag.

Apart from the addresses in Etherscan "Heist" list, we also collect other incident information for a more complete analysis, such as incident type,[5] incident cause, stolen amount,[6] occurrence year,[7] etc. (See detailed analysis in Section IV-B)

Discussion of Our Incident Searching: It is essential to accurately identify the source addresses of money laundering cases. These addresses provide the data foundation for tracking downstream transactions, which is also true for AML on other blockchain platforms with account-based models similar to Ethereum. These source addresses for money laundering cases are also addresses of attacking hackers or fraudsters. Collecting these source addresses can also facilitate research on attack defense and anti-fraud for the blockchain academic community. It should be noted that the statistics of Etherscan only account for incidents from "crypto-native" crimes (i.e., on-chain crimes), in which illicit profits are almost always obtained in the form of cryptocurrency rather than fiat currency.

B. Track Diverse Downstream Transactions

After obtaining the placement address set of incidents, the next natural step is to develop a complete tracking and collection of the hackers' downstream layering transactions, to identify the layering and integration address set.

In our downstream AML tracking, we have to deal with the challenge of *large and sparse blockchain transaction data*. Due to the unlimited number of blockchain account openings and transactions, cybercriminals can create a large number of pseudonymous accounts and transactions at near-zero cost, resulting in a huge volume of data and sparse correlations for their interactions. Such a large downstream money laundering chain is not conducive for investigating the crimes.

To address this challenge, we first investigate and characterize the labeled layering addresses of a real incident – Upbit Hack. Then, inspired by the insights of this case, we employ the idea of taint analysis to develop an AML tracking algorithm.

1) Case Study of Upbit Hack Money Laundering: Before designing the tracking algorithm, it is necessary to understand the transaction features of crypto ML accounts as a basis. Although blockchain data is publicly available, and also the source of money laundering is available through victim project

[5]The incident type is categorized based on the type of object attacked.

[6]The incident cause and stolen amount are collected from media reports or official statements of the incident.

[7]The occurrence year is determined based on the year of the first transaction at the address of the hacker's crime.

TABLE I
THE NUMBER OF ADDRESSES IN EACH LAYER IN THE UPBIT HACK ML PROCEDURE

Level	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Number	3	1	3	6	26	23	91	155	122	190	147	27	11	5	3	2

TABLE II
THE TRANSACTION CHARACTERISTICS OF THESE 815 ADDRESSES FLAGGED AS "UPBIT HACK". "TX" IS SHORT FOR TRANSACTION AND"#" DENOTES THE NUMBER

	#Outgoing-Tx	#Incoming-Tx	#All-Tx
Max	533	105	603
Median	1	2	3
Min	0	1	1
Average	5.81	3.50	9.31

Fig. 6. Examples of two typical Bitcoin taint analysis policies, Poison and Haircut, where nodes denote BTC transactions and edges denote the flow of BTC values between two transactions. The black nodes denote the initial blacklisted transaction.

reports, it is still difficult to find open Ethereum ML datasets for reference. Fortunately, while we are collecting 115 source addresses for Phase I on Etherscan, the set of money laundering addresses is found associated with the hacker address with the name tag called "Upbit Hack". This incident occurred in November 2019 in the South Korean exchange called Upbit, where unidentified attackers stole 342,000 ETH from the exchange's hot wallet and transferred it to an unknown wallet address with a total value of nearly $45 million.

In the end, a total of 815 external accounts are flagged on Etherscan as being related to the Upbit attack. Etherscan indicates the *level* of ML by the number of hops from the original placement account. The address of the hacker who directly steals funds from the Upbit exchange is labeled "Hacker 1" (it is one of the original 115 addresses that were flagged as "Heist" in Phase I). Subsequently, deeper levels are labeled by the number of hops and time, e.g., the accounts of the next level from Hacker 1 are labeled "Hacker 2.1", "Hacker 2.2" and so on. We first count the number of money laundering layers in this Upbit Hack case, as shown in the TABLE I. Then, we calculate the transaction characteristics of these 815 laundered addresses, as shown in the TABLE II. Based on these analyses and results, we obtain the following findings:

- **Finding 1: number of laundering layers.** The number of money laundering layers in this case is as high as 15, which far exceeds the existing experimental settings of AML in traditional finance with 3–5 layers [29].
- **Finding 2: number of addresses per layer.** For the number of addresses at different levels, TABLE I shows a pattern of "more in the middle, two short less", and the largest address number is 190, accounting for nearly a quarter of all "Upbit Hack" addresses. Although opening new accounts is almost free, it also increases the time cost for criminals to recover these dispersed funds during the aggregation phase.
- **Finding 3: transaction number of laundering addresses.** The number of transactions in the money laundering accounts is not large, the maximum total number of transactions is 603 (less than 1000), the average number is 9.3, and the median number is even only 3. This is also in line with the characteristic of "diversification of asset" to avoid a transaction of too large an amount to attract the attention of the blockchain whale monitoring system.

These findings provided the basis for the subsequent design of the money laundering tracking algorithm.

2) Proposed AML Tracking Algorithm: Inspired by the findings provided by Upbit Hack, we propose the first AML tracking algorithm in the Ethereum scenario.

Our basic idea of the AML tracking algorithm is taint analysis of money flow [5]. Two typical policies for taint analysis in the Bitcoin scenario are Poison and Haircut [30]. Fig. 6 shows the examples for these two typical taint analysis policies. In the former policy, the receivers of the blacklisted transactions are all considered to be tainted as long as the transaction have at least one dirty incoming transaction, as shown in Fig. 6(a). In the latter policy, by taking the amount value of the dirty incoming transaction into consideration, each outgoing transaction contains the proportion of the dirty and clean input value, as shown in Fig. 6(b). However, these two policies for UTXO model-based Bitcoin are not suitable for account model-based Ethereum, where the Ethereum address can be reused like a bank account.

To trace ML addresses in the Ethereum, in this work, we design the Truncated Poison Policy (TPP), where the receiver of a blacklisted account is considered to be suspicious unless it behaves like an ordinary innocent user or is labeled as a known service provider. Formally, the entire set of Ethereum transaction records is modeled as a transaction network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ is the set of all Ethereum addresses. $\mathcal{E}$ is the set of all Ethereum transactions, where each edge is a directed edge, and (u, v) refers to sending an asset from address u to v. For a given incident, let Cur_k be the set of suspicious ML accounts in the current k-th layer. With TPP, the set of candidate ML accounts in the next layer is obtained as

$$Cur_{k+1} = \{v|(u, v) \in E, u \in Cur_k\} - \{v|v \in SP \vee User(v)\},$$

where SP refers to the set of service providers' addresses or the set of suspected addresses that provide the service, and the $User(v)$ function refers to the fact that v is behaving

like an ordinary user (detailed described in the AML tracking algorithm below).

The pseudo-code of the TPP-based AML tracking algorithm is as shown in Algorithm 1. Our algorithm collects ML-related transaction records layer by layer and finally obtains a hierarchical set with several layers. The iterative collection process of each incident contains the initialization step, loop step, and termination step:

• **Initialization step.** AML tracking starts with the placement address set (See more details in Section III-A). Put the source address of the heist collected in Phase I to the placement address set $\mathcal{P}$ (line 1). Also, set the current layer level k to 0 (line 2).

• **Loop step.** For any address a in the current address set Cur_k at layer k, we have following steps:

1) **Transaction query.** Apply an on-chain data crawling method to query its external, internal, and ERC20 transactions, and get the transaction record $\mathcal{T}_a$ (`QuaryTxns()` in line 5). We will introduce more about our data crawling in data collection in Section IV-A.1.
2) **Calculate the dirty amount.** Filtering out transactions containing small amounts of dirty money ($\leq$ threshold β, $\beta = 0.01$ for Upbit Hack) (`DirtyAmount()` in lines 6). The purpose of money laundering is to conceal the origin of illicit funds and thus the process tends to be very low profile and avoids using one address for a large number of transactions. Therefore, ML usually involves intensive and non-small-amount transactions between a group of accounts.
3) **Drop unknown service addresses.** We consider the address a with a large number of transactions ($|T_a| > \Omega$, Ω is the threshold) to be an unknown service address in the aggregation phase rather than the layering phase (lines 7–9). We set the threshold Ω to be 1000, according to the **Finding 3** in the Upbit Hack Case. We then retain the transactions between unknown service providers and upstream laundering accounts $\bigcup_1^k Cur_i$ within one week as suspected money laundering transactions (`FilterTxns()` in line 10).
4) **Add ML layering address label for the current layer.** Tag the remaining address a with 'Heist' label in the Address Label Library Lib, and record the current level k it is in (line 12).
5) **Select candidate ML address for next layer.** For address a with the 'Heist' label, we select the next level $k+1$ of suspicious addresses Cur_{k+1} from recipient addresses of a's outgoing transactions. Also, we have a timing constraint, i.e., the next layer of suspicious addresses needs to be after the time when the dirty money is transferred from the current layer of addresses. Then, we check whether these recipient addresses are unmarked addresses (`GetUnmarked()` in line 13), according to the address label library Lib (The label library will be described in detail in Section IV-A.2). Addresses that meet the aforementioned criteria will be added to downstream candidate ML addresses for the next layer Cur_{k+1}. We also select the addresses of known service providers from recipient addresses of a's outgoing transactions (`GetServices()` in line 14) and add them to $\mathcal{I}$.
Then, the transactions of address a are added to the transaction set $\mathcal{T}$ (line 15).

Algorithm 1 AML Tracking Algorithm

Data: placement address set $\mathcal{P}$, address label library Lib
Input: max. depth of traced layers K, max. number of addresses per layer Ψ, threshold transaction number for unknown services Ω
Result: layering address set $\mathcal{L}$, integration address set $\mathcal{I}$, involved transaction set $\mathcal{T}$

1 $k \leftarrow 0$; `// The tracing depth`
2 $Cur_k \leftarrow \mathcal{P}$; `// The current suspicious address set`
3 **while** $k \leq K$ *and* $0 < |Cur_k| < \Psi$ **do**
4 **for** *address* $a \in Cur_k$ **do**
5 $\mathcal{T}_a \leftarrow$ `QuaryTxns(a)`;
6 **if** `DirtyAmount` $(\mathcal{T}_a, \bigcup_{i=0}^k Cur_i) > \beta$ **then**
7 **if** $|T_a| > \Omega$ **then**
8 $\mathcal{I} \leftarrow \mathcal{I} \cup \{a\}$;
9 $Cur_k \leftarrow Cur_k - \{a\}$;
10 $\mathcal{T}_a \leftarrow$ `FilterTxns` $(T_a, \bigcup_{i=0}^k Cur_i)$;
11 **else**
12 `AddItem` $(Lib, a,$ 'Heist', $k)$;
13 $Cur_{k+1} \leftarrow Cur_{k+1} \cup$ `GetUnmarked` $(\mathcal{T}_a, Lib)$;
14 $\mathcal{I} \leftarrow \mathcal{I} \cup$ `GetServices` $(\mathcal{T}_a, Lib)$;
15 **end**
16 $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{T}_a$;
17 **end**
18 **end**
19 $k \leftarrow k + 1$;
20 **end**
21 $\mathcal{L} \leftarrow \bigcup_{i=1}^k Cur_i$;

• **Termination step.** We keep increasing the depth k (line 19) until the depth exceeds the maximum number of layers K, or the size of the current addresses set Cur_k exceeds the range $[0, \Psi]$ (line 3). Here we choose conservative parameters $K = 20$ according to Finding 1, $\Psi = 10,000$ according to Finding 2. Finally, we merge the addresses of each layer to obtain the final layering address set $\mathcal{L}$ (line 21).

Discussion of Our AML Tracking Algorithm: The tracing parameters, including the maximum depth of traced layers K, the maximum number of addresses per layer Ψ, and the threshold transaction number for unknown services Ω, can be adjusted during the actual application. The purpose of setting these parameters is to control the scope of transaction tracing, to avoid an explosion in the number of downstream addresses, and to delineate the conditions for terminating tracing. More explanation of the performance of the proposed AML tracking algorithm will be discussed in experimental results and analysis in Section IV-B.

IV. Experimental Results and Analysis

In this section, we conduct experiments and analysis on several Ethereum representative incidents. By applying the proposed crypto ML identification algorithm (cf. Section III), we acquire the corresponding placement (Phase I), layering (Phase II), and integration (Phase III) address set for each incident. Based on these collected datasets, we systematically conduct feature analysis and evolution analysis on these three main phases of ML. Several interesting and novel insights will be demystified and contribute to further crypto AML research.

A. Data Collection

1) On-Chain Data Crawling: In Section III-B, we mention that one crucial step for ML identification is to query the Ethereum transactions of a downstream address from heist addresses. Here we introduce our transaction acquisition method for crypto ML identification.

Although existing data analysis work uses the method of synchronizing the Ethereum block data (e.g., [31]), this method is not applicable to our study. As discussed in [32], the reason is mainly that the block data is sorted by time (i.e., time-intensive data), but what we need is to crawl the transaction records of a given source address (i.e., space-intensive data [31]). Finding all transactions for a given address from the block data is much more time-consuming, whereas, the official Ethereum browser (i.e., Etherscan) provides APIs to extract all the participating transactions based on the address. Therefore, the API-based data construction method is more appropriate in crypto AML.

Specifically, leveraging Etherscan's APIs, we develop a transaction crawler with the breadth-first search (BFS) policy. The BFS policy starts from a specified address and crawls all its one-hop incoming and outgoing transaction records, including external transactions, internal transactions, and ERC20 transactions. In this work, with the BFS policy, all the one-hop downstream transactions of heist addresses can be tracked and checked as completely as possible with no leakages. More searching policies of this transaction crawler have been proposed in our previous research [33], [34].

Besides, the transaction records we obtained in XBlockFlow include the following information: the `hash` field denotes the unique identifier of the blockchain transaction, the `from` and `to` field denotes the sender and receiver, the `timeStamp` field denotes the UNIX timestamp of transaction time, the `blockNumber` field denotes the block number that the transaction is in, the `tokenSymbol` and `contractAddress` field denotes the symbol and contract address of transferred token, the `isError` denotes the status of the transaction, the `gasPrice` and `gasUsed` field denotes the price and utilization of gas fee[8] on Ethereum.

2) Off-Chain Address Label Library: In Section III-B, we mention that the label library is denoted as Lib and here we explain more about the label library in our experiments. Address labels are important references to accomplish the auto AML tracking in this work because these address labels guide the AML algorithm when to stop tracking. Our labels consist of two parts: the labels of the service providers and token contracts. To determine the service providers, we crawl label addresses associated with exchanges (e.g. "Exchange", "DEX", etc.), mixing services (e.g. "Tornado.Cash"), and other label addresses that appear in connection with actual money laundering activities. We obtain more than 260,000 items, which is sufficient to cover money laundering destinations. The labels of the service providers are open on our website XBlock (https://xblock.pro).[9] The token contracts labels refer to the "ERC20TokenInfo" dataset with more than 313,000 ERC20 tokens, and the "ERC721TokenInfo" dataset with more than 15,000 ERC721 tokens, which are published in our previous research [35], including contract addresses, token names, token symbols, etc. These datasets can help identify the types of tokens being used for money laundering in token transactions.

B. Results and Analysis of Placement (Phase I)

In this part, we explain how we select representative incidents for experiments, and demonstrate the statistics of the identified placement address set of each incident.

As mentioned earlier in Section III-A, Etherscan's "Heist" list has 115 addresses of known security incidents, including tagged accounts of hackers, exploiters, and scammers. As shown in Fig. 5, each address in Etherscan's "Heist" list has a name tag, indicating the associated incident. We model each address as the corresponding placement set $\mathcal{P}$ for each incident. The list of 115 addresses and corresponding incidents is shown in our supplementary material [19].

1) Statistics on the Number of Incidents: An incident may have more than one address tagged with "Heist" on the Etherscan. First, we cluster the addresses into individual incidents by name tag. We exclude one address for which Etherscan does not provide a name tag (because then we do not know what case it belongs to), and aggregate the hackers, exploiters, and scammers belonging to the same case together, and finally, we get 73 incidents.

2) Statistics on the Distribution of Incidents by Year and the Amount Stolen: Next, we count the years of the occurrence of these 73 incidents, and the results are displayed in TABLE IV. The year of the case is based on the year of the first transaction at the address of the hacker's crime. TABLE IV shows that the distribution of the number of cases each year is quite uneven. In the early stages of Ethereum development, there was only one recorded case per year in 2016 and 2017. In the subsequent years, the number increased annually, reaching its peak in 2021. At this time, with the rise of DeFi but various DeFi projects still in their infancy, 67% of the 49 security incidents that occurred in 2021 were DeFi exploits.

To this end, we gather a comprehensive dataset comprising 73 security incidents that took place within the Ethereum ecosystem between 2016 and 2022. The detailed basic information of these incidents can be found in TABLE III. Our dataset encompasses four primary categories of security incidents: CEX hacks, DeFi exploits, scams, and others. The

[8]Users are required to pay a transaction fee for each transaction conducted on Ethereum.

[9]XBlock collects the current mainstream blockchain data and is one of the blockchain data platforms in the academic community.

TABLE III
THE 73 ETHEREUM INCIDENTS FROM 2016-2022 LABELED BY ETHERSCAN AND THEIR BASIC INFORMATION. $|\mathcal{P}|$ DENOTES THE SIZE OF PLACEMENT ADDRESS SET

Case Name	Case Type	Year	Amount Stolen	$\|\mathcal{P}\|$	Case Name	Case Type	Year	Amount Stolen	$\|\mathcal{P}\|$
GatecoinHacker	CEX Hack	2016	$2.14M	4	NowSwapExploiter	DeFi Exploit	2021	$1M	1
MultisigExploitHacker	Others	2017	$30M	1	PancakeHunnyExploiter	DeFi Exploit	2021	$2M	1
CoinrailHacker	CEX Hack	2018	$40M	1	PolyNetworkExploiter	DeFi Exploit	2021	$600M	1
BancorHacker	DeFi Exploit	2018	$23.5M	1	PopsicleSwapHacker	DeFi Exploit	2021	$25M	2
SpankChainHacker	Others	2018	$40K	1	PunkProtocolExploiter	DeFi Exploit	2021	$3.95M	1
FakeMetadiumPresale	Scam	2018	-	1	RariCapitalExploiter	DeFi Exploit	2021	$15M	1
BitpointHacker	CEX Hack	2019	$32M	2	SashimiSwapExploiter	DeFi Exploit	2021	$210K	1
CryptopiaHacker	CEX Hack	2019	$45K	3	SirenProtocolExploiter	DeFi Exploit	2021	$2.85M	4
DragonExHacker	CEX Hack	2019	-	1	THORChainExploiter	DeFi Exploit	2021	$13M	4
UpbitHacker	CEX Hack	2019	$2B	1	UraniumFinanceHacker	DeFi Exploit	2021	$50M	1
PlusTokenPonzi	Scam	2019	$49M	1	VeeFinanceExploiter	DeFi Exploit	2021	$35M	1
KucoinHacker	CEX Hack	2020	$280M	1	VesperFinanceVUSDExploiter	DeFi Exploit	2021	$1M	2
AkropolisHacker	DeFi Exploit	2020	$2M	1	VisorFinanceExploiter	DeFi Exploit	2021	$8.1M	1
HarvestFinanceExploiter	DeFi Exploit	2020	$24M	2	WaultFinanceExploiter	DeFi Exploit	2021	$800K	1
Lendf.MeHacker	DeFi Exploit	2020	$25M	1	xTokenExploiter	DeFi Exploit	2021	$245M	1
WarpFinanceHacker	DeFi Exploit	2020	$7.7M	1	Yearn(yDai)Exploiter	DeFi Exploit	2021	$11M	1
NexusMutualHacker	Scam	2020	$8M	1	BadgerDAOExploitFunder	Others	2021	$120M	1
AscendEXHacker	CEX Hack	2021	$78M	2	DAOMakerExploiter	Others	2021	$7M	3
BitmartHacker	CEX Hack	2021	$150M	2	MisoFrontEndExploiter	Others	2021	$3M	1
LiquidExchangeHacker	CEX Hack	2021	$80M	1	VulcanForged	Others	2021	$140M	2
AlphaHomoraV2Exploiter	DeFi Exploit	2021	$37M	1	0xhabitatethMultisigDrainer	Scam	2021	$123K	1
AnySwapV3Hack	DeFi Exploit	2021	$78.7M	1	AFKSystem	Scam	2021	$8M	1
BentFinanceExploiter	DeFi Exploit	2021	$1.8K	3	AnubisDAOLiquidityRug	Scam	2021	$60M	3
BondlyFinanceExploiter	DeFi Exploit	2021	$200M	1	BELLEHoneypotRugPull	Scam	2021	-	1
BrincFinanceExploiter	DeFi Exploit	2021	$1.1M	1	EtherwrappedRugContract	Scam	2021	$12M	2
BunnyFinanceExploiter	DeFi Exploit	2021	$200M	1	MetaDAORug	Scam	2021	$3.2M	1
bZxPrivKeyExploiter	DeFi Exploit	2021	$55M	5	MommyMilkersHoneypotToken	Scam	2021	-	1
ChainPortExploiter	DeFi Exploit	2021	-	5	RUNETokenExploiter	Scam	2021	$76K	1
CreamFinanceExploiter	DeFi Exploit	2021	$130M	4	SaturnbeamFiRugPull	Scam	2021	$10M	1
EasyfiHacker	DeFi Exploit	2021	$80M	1	ATOStolenFunds	CEX Hack	2022	$1.5M	1
FinNexusHacker	DeFi Exploit	2021	$11.2M	1	LCXHacker	CEX Hack	2022	$6.8M	1
ForceVaultHacker	DeFi Exploit	2021	$376K	2	CashioAppExploiter	DeFi Exploit	2022	$52.8M	1
FurucomboHacker	DeFi Exploit	2021	$14M	2	FloatProtocolFuseExploiter	DeFi Exploit	2022	$250K	1
ImpossibleFinanceExploiter	DeFi Exploit	2021	$500K	1	DEGOandCocosExploiter	Others	2022	$10B	1
IndexedFinanceExploiter	DeFi Exploit	2021	$16M	1	Arthur0xWalletHacker	Scam	2022	$1.6M	1
MaliciousActorExploitAbuser	DeFi Exploit	2021	$589K	2	Fake Phishing5041	Scam	2022	$91M	1
MonoXFinanceExploiter	DeFi Exploit	2021	$31M	2					

TABLE IV
THE DISTRIBUTION OF CASES BY YEAR

Year	2016	2017	2018	2019	2020	2021	2022	Sum
#Cases	1	1	4	5	6	49	7	73

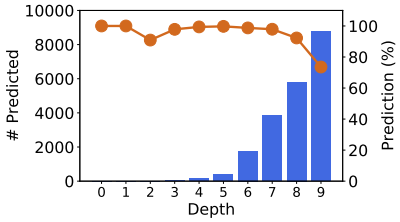


Fig. 7. Experiments to evaluate the AML Transaction Tracing Algorithm for Upbit Hack case.

"others" category includes incidents such as exploits related to Decentralized Autonomous Organizations (DAOs), Game Finance (GameFi), Non-Fungible Token Finance (NFTFi), and so on. For more comprehensive information regarding the incidents, we refer the readers to our supplemental material [19].

C. Results and Analysis of Layering (Phase II)

Based on the placement address set, we determine and evaluate the layering addresses associated with each incident by applying the proposed AML tracking algorithm. Subsequently, we employ statistical and analysis techniques of the selected representative incidents to delve into and comprehend the layering addresses involved in crypto money laundering.

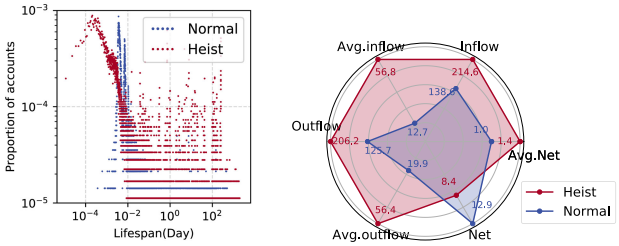
1) Performance of Layering Address Set Identification: To assess the accuracy of identifying the layering address set, we conducted a validation experiment using Upbit Hack dataset with partially marked layering accounts, as mentioned in Section III-B. Based on the Upbit Hack placement address set identified in the previous part, we employed Algorithm 1 to identify suspicious Upbit Hack layering addresses. We calculate precision at various tracing depths, as depicted in Fig. 7. The results indicate that as the tracing depth increases, the number of detected layering addresses grows exponentially. Remarkably, even at a depth of 8, the precision remains consistently above 90%. At the same time, we calculate the recall of Algorithm 1 on the Upbit Hack dataset, and the result is 96.2%. This means that the vast majority of money laundering addresses labeled as "Upbit Hack" by Etherscan can be identified by our algorithm.

However, practical applications of our algorithm may encounter false positives and false negatives due to the absence of true and complete labeling in the public domain. False negatives occur when a layering address associated with a

heist surpasses the threshold Ω, leading it to be classified as an unknown service and subsequently excluded. Increasing the threshold Ω can reduce the false negative rate, but it comes at the cost of additional computations and potentially higher false positive rates. False positives manifest in two scenarios. The first scenario involves an on-chain-off-chain transaction between the cybercriminal and an unwitting user. The user unknowingly sends legitimate off-chain assets, such as fiat cash, to the criminal while the criminal transfers the on-chain stolen asset to the user. Our AML tracking algorithm labels such user addresses as layering addresses because these accounts receive dirty money from criminals. Cases involving off-chain transactions inevitably result in false positives within our research and cannot be entirely eliminated. The second scenario relates to the lack of updates to the label library Lib. If an outgoing address from a "Heist" address belongs to a new service provider or a new token project, but the service provider is not included in the updated label library, the algorithm could mistakenly implicate the corresponding service provider. Regularly updating the label library can mitigate this kind of false positive.

2) Overview Statistic of Layering: We conduct preliminary data analysis and exploration of the layering address sets and transaction sets of representative incidents, providing the following insights (refer to our supplemental material [19] for the complete table). **(i) In terms of duration,** these incidents varies from less than 1 day to approximately 3 years. Notably, incidents occurring in the earlier years tend to have longer durations. For instance, the Upbit Hack laundering lasted for over 2 years, whereas cases in 2022 lasted as short as one day. This observation may be attributed to the utilization of newer methods in the Ethereum ecosystem, such as mixing services. An example is the LCX exchange hack that occurred in 2022, where the hacker only required approximately a day to exchange the stolen ERC20 tokens for Ether through a decentralized exchange (DEX) and eventually transferred them to a mixing service named Tornado.Cash. **(ii) In terms of the monetary value involved,** the average amount laundered in these cases ranges from \$100,000 to \$1 million, with the highest recorded value reaching \$10 million. This underscores the significant financial losses associated with these incidents. **(iii) In terms of the complexity of the cases,** we analyze three key aspects: the number of layers (tracing depth), transaction fees (gas cost), and the size of the transaction set $\mathcal{T}$ for each incident. Typically, incidents involving more layers of money laundering exhibit a greater number of accounts in $\mathcal{L}$ and transactions in $\mathcal{T}$, resulting in larger and more intricate transaction data. This complexity facilitates the ability of hackers to hide and obscure the origin of stolen funds, but also entails higher transaction fees for the hackers.

3) Understanding Layering via Graph Analysis: Given the previous studies that have examined network-based measurements and investigations on the Ethereum blockchain [31], our objective is to conduct a graph analysis of transactions associated with crypto ML groups. This analysis aims to uncover the distinctions between ML layering transaction networks and the normal transaction network.



(a) Lifespan distribution (b) Transaction amount (ETH)

Fig. 8. Comparison of node trading characteristics of transaction graphs.

To accomplish this, we begin by modeling the layering transactions of each incident as a network, denoted as $G = (V, E)$. Here, E represents the edge set comprising all transactions in the incident (i.e., $\mathcal{T}$), and V represents the node set, which signifies the accounts involved in these transactions.

For each incident, we establish two networks: HeistEthNet for the Ether transactions (including external and internal transactions) and HeistTokenNet for the ERC20 token transactions. These networks are intended for comparison with TransactionNet and TokenNet of the previous research [31], respectively. It is important to note that the entire network [31] provides insights into the characteristics of the normal transaction network since our money laundering accounts account for a minuscule 0.3% of the entire transaction network (46 million transactions)

4) Global Network Properties: When examining the global network properties of layering, we focus on three key aspects: lifespan, transaction amounts, and density.

a) Lifespan: Fig. 8(a) demonstrates two noteworthy observations. Firstly, layering accounts associated with heists often have extremely short lifespans, indicating a "used-and-dumped" characteristic. The lifespan distribution for layering accounts of heists skews towards shorter durations compared to normal accounts. Secondly, layering accounts with longer lifespans exhibit an irregularly high percentage of jumps. This can be attributed to cautious hackers who delay the transfer of stolen funds, waiting for an opportune moment to conduct the laundering process. For instance, the Coinrail hacker[10] stole assets in 2018 and remained dormant for two years until 2020 when the stolen funds were finally transferred out.

b) Transaction amount: We analyze the inflow, outflow, and net value of layering accounts for each heist, along with the corresponding average value per transaction of these accounts, as depicted in Fig. 8(b). The findings reveal that the transaction amount of layering accounts is significantly larger than those of normal accounts across almost all indicators. This indicates that despite the ability of hackers to create accounts without restrictions, the amount of stolen funds is so large that the amount per transaction in the laundering process remains large. The average inflow and outflow values of layering accounts exceed 50 ETH, approximately 3-5 times higher than those of normal accounts. The lower net value of layering can be attributed to the practice of minimizing the

[10]https://cn.etherscan.com/address/0xf6884686a999f5ae6c1af03db92bab9c6d7dc8de

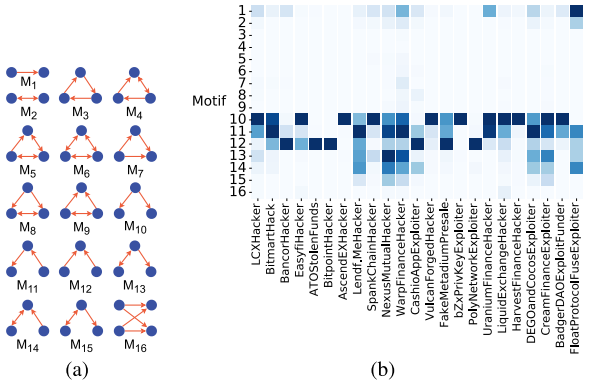


Fig. 9. (a) Directed motifs: M_1 and M_2 are all connected two-node motifs; M_3 – M_{15} are all 13 connected three-node motifs; M_{16} is the four-node bi-fan motif. (b) Distribution of the various motifs in money laundering networks.

TABLE V
COMPARISON OF NETWORK PROPERTIES. ("MED." MEANS MEDIAN. "AVG." MEANS AVERAGE)

	Self-loop	Reciprocity	Density (s.,undi)[11]	Density (multidi)	Global cluster	Avg. pathlen
HeistEthNet (Med.)	0.00%	7.85E-02	1.25E-01	3.69E-01	1.35E-02	1.85
HeistEthNet (Avg.)	0.55%	1.21E-01	**4.80E-01**	**1.70E+00**	2.96E-02	3.85
TransactionNet [32]	0.13%	3.00E-02	1.24E-07	1.87E-07	1.00E-01	5.33
HeistTokenNet (Med.)	0.00%	4.14E-02	1.25E-01	1.70E-01	2.44E-03	1.33
HeistTokenNet (Avg.)	0.02%	1.05E-01	2.56E-01	4.92E-01	1.30E-02	3.41
TokenNet [32]	0.19%	3.00E-02	2.03E-07	1.87E-07	1.75E-01	3.87

amount of money left in layering accounts, aiming to reduce the risk of freezing.

c) Density: We evaluate the network density based on established formulas from prior research [31]. The results, as presented in TABLE V, provide interesting insights. Both HeistEthNet and HeistTokenNet exhibit a density of twice the average density in multidigraph compared to simple undirected graph, with HeistEthNet even reaching 3.5 times. In contrast, the density of TransactionNet in the entire network shows only a 1.5-fold increase in multidigraph compared to the simple undirected graph. These findings indicate that crypto money laundering networks form frequent and densely interconnected subnetworks. More explanation of TABLE V is given in our supplementary material [19].

5) Local Network Properties: Moving on to the analysis of local network properties in layering, we focus on two key aspects: the self-loop ratio and network motifs.

a) Self-loop ratio: Self-loop transactions, where the sender and receiver addresses are the same, are not in line with the objective of money laundering, as they do not involve splitting and diverting funds. As expected, the self-loop ratio in HeistTokenNet is lower compared to TokenNet. However, interestingly, the average self-loop ratio in HeistEthNet is higher than that in TransactionNet. Further investigation reveals that this difference is primarily due to the CashioApp Exploiter,[12] who left messages to the community through self-transactions, resulting in a high self-loop ratio in this incident.

[11]Simple, undirected graph.

[12]https://etherscan.io/address/0×86766247ba3405c5f15f06b895294200809e9cfb

TABLE VI
TOP TEN SERVICE PROVIDERS

	Name tag of Address	Discription
1	Uniswap V2: Router 2	Proxy contract for handling trades and fund flows of Uniswap Dex
2	Tether USD	A kind of stablecoin that maintains a 1:1 ratio with the US dollar
3	Uniswap V3: Router 2	Proxy contract for handling trades and fund flows of Uniswap Dex
4	Binance	A centralized exchange
5	Uniswap V2	The second version of the Uniswap protocol
6	Binance 14	A wallet of Binance exchange
7	USD Coin	A kind of stablecoin that maintains a 1:1 ratio with the US dollar
8	OpenSea	An NFT trading platform
9	EtherDelta 2	A wallet of EtherDelta exchange
10	Wrapped Ether	Ether being packaged within the ERC-20 token standard

b) Network motifs: Network motifs capture higher-order structural patterns within a network, representing recurring subgraphs [36]. To identify these patterns, we examine the percentage of directed motifs in the simple directed money laundering network for each incident. Fig. 9 presents the results for the motif percentages in 23 incidents, while encountering out-of-memory errors for the remaining cases. We then compare these findings with the entire Ethereum blockchain network [31], yielding interesting insights. Notably, crypto ML networks exhibit significantly lower fractions of *closed triangular motifs* (M_3–M_9). This could be attributed to the fact that closed triangular motifs often signify internal asset circulation, such as wash trading behavior [37], which is inconsistent with the objective of ML. Conversely, *open triangle motifs* are the most frequent motifs in ML networks, i.e., M_{10}-M_{12}. These three motifs correspond exactly to three phases of ML: M_{10} corresponds to the placement phase, spreading the illegally obtained funds and extending the money trail; M_{11} corresponds the layering phase, continuously transferring stolen funds to complicate traceability; and M_{12} corresponds the aggregation phase, consolidating scattered laundered funds for withdrawal.

It is worth noting that the networks of DeFi exploit incidents display a higher occurrence of M_{13}-M_{15} motifs, where bidirectional edges are most likely associated with a typical DeFi action - token swap (i.e., exchange). This DeFi action is related to the integration phase of ML, which will be discussed in more detail in Section IV-D.2.

D. Results and Analysis of Integration (Phase III)

Continuing our investigation into the different phases of crypto ML, we now delve into the integration phase. This section presents the results and analysis of the integration phase, focusing on the service providers of laundering exit as well as advanced obfuscation methods.

1) Service Providers of Laundering Exit: Following crypto security incidents, it is common for stolen funds to flow through various service providers for laundering. Thus, it is crucial to present the percentage and changes of various service providers.

In TABLE VI, we conduct a study on the top 10 service providers in terms of transaction volume in the integration phase. We observe that three of these providers are DEXs, Uniswap. This indicates that illicit funds are being utilized for token exchange. Additionally, we identify three prominent tokens – Tether USD, USD Coin, and Wrapped Ether – that

TABLE VII
PART OF THE MAPPING TABLE OF SERVICE PROVIDERS TO A CATEGORY

Service Provider	Category	Reference Website
Binance	Centralized Exchanges	https://www.binance.com/en
Uniswap	Decentralized Exchanges	https://uniswap.org/
Polygon	Crossing Chain Services	https://wiki.polygon.technology/docs
Aave: Lending Pool	Loan Services	https://docs.aave.com/
Tornado.Cash	Mixing Services	https://en.wikipedia.org/wiki/Tornado_Cash
MEV Bot	Other	https://mevboteth.com/

are frequently involved in these transactions. This implies that a significant portion of the illicit funds is converted into stablecoins or Wrapped Ether, possibly for interoperability with other blockchain networks. Noteworthy service providers include the well-known CEX, Binance, and the DEX, EtherDelta. Furthermore, our analysis uncovers evidence of illicit funds being channeled into the OpenSea platform, indicating hackers' utilization of black money for the purchase of NFT tokens.

To further analyze the evolution of service providers involved in crypto ML over time, we categorize them into six groups: centralized exchanges (CEXs), decentralized exchanges (DEXs), crossing chain services, loan services, mixing services, and others. The mapping of service providers to these categories is provided in Table VII (a portion of the mapping is shown here, while the complete contents can be found in our supplemental material [19]).

In Fig. 10, we present a stacked bar chart that illustrates the changing percentages of different service providers over time. The chart reveals cybercriminals shifting preferences in the destination of stolen funds. Centralized exchanges, which were the primary choice for laundering money in 2018, experienced a decline in 2019. This could be attributed to stricter anti-money laundering (AML) and know-your-customer (KYC) procedures implemented by CEXs in response to regulatory requirements [38], [39]. Conversely, the share of DEXs increased, suggesting that DEXs without a centralized third party are more likely to evade law enforcement investigations. Furthermore, the percentage of crossing chain services has been steadily growing since 2019. Crossing chain services enable dirty money to circulate and obfuscate across multiple blockchain networks, indicating the increasing sophistication of criminals. Notably, there is a flow of dirty money into lending services such as Aave, Compound, and Dydx. Criminals leverage the liquidity pools of these lending services to conceal the origin of illicit funds, reduce traceability, and even earn additional income by providing large amounts of dirty money to the pools. Additionally, Tornado.Cash, a mixing service introduced in 2019, has become a popular destination for money laundering. Its effectiveness as a classic laundering service is evident, as Tornado.Cash has recently faced sanctions from the U.S. Office of Foreign Assets Control (OFAC).[13] Lastly, there are other types of service providers. For example, criminals deposit crypto to Air Wallet, which functions as a distributed airdrop and digital wallet platform.

2) Advanced Obfuscation Methods: Having examined the various service providers involved in money laundering exit strategies, we now shift our focus to exploring the advanced obfuscation methods employed during the integration phase.

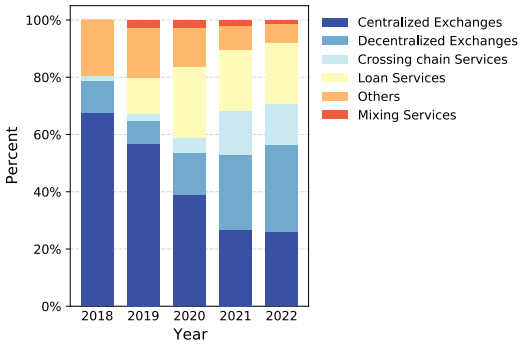


Fig. 10. Evolution of destination service providers from 2018 to 2022.

a) Method I: DEX token swap: As discussed in Section IV-C.5, DeFi exploit incidents often involve frequent token swap actions with DEXs. DeFi token swap refers to a trader's account selling a certain amount of a certain token in exchange for a certain amount of another token in a liquidity pool of an Automated Market Maker (AMM). In order to analyze these token swaps, we propose Algorithm 2 to identify and count the occurrences of token swaps in each incident's transaction set $\mathcal{T}$. Typically, a token swap consists of two transfers involving ETH or tokens, which are commonly completed within a single transaction. For a transaction to be considered a token swap, two non-zero transfers must occur between two accounts with different asset types and opposite directions. Note that neither the sender nor the receiver of these transfers should be the zero address, as this implies an act of minting tokens and an act of burning tokens, respectively. For evaluation, we input the transaction hashes of the identified swap pairs into Etherscan and obtain the "Transaction Action" field or "Method" field (the parsed function name from input data) of the transaction. If Etherscan returns fields include "Swap," "Exchange," or "Trade," it indicates that the identified transfer pairs represent the semantics of a swap, confirming the accuracy of the prediction.

We provide more evidence to support the observation that ML behaviors after DeFi exploit incidents frequently involve a high occurrence of token swap actions. For instance, 70 token swaps are identified in money laundering of Cream Finance Exploiter,[14] and its M_{13}-M_{15} motif fractions are also relatively high referring to Fig. 9. To gain further insights, we explore and analyze the token swap behavior in the crypto ML process. We aim to understand the purpose behind these activities and attribute them to the following actions:

- *Swapping tokens to non-freezable assets.* For example, Tether (USDT) is a stablecoin pegged to the US Dollar, operated by Tether Limited Inc. USDT issuers may freeze assets held by illegal addresses. As a result, criminals use DEXs to swap freezable assets for non-freezable ones. For example, in the AscendEX Exploit[15] event

[13]https://home.treasury.gov/news/press-releases/jy0916

[14]https://etherscan.io/address/0×24354d31bc9d90f62fe5f2454709c32049cf866b

[15]https://etherscan.io/address/0 × 2c6900b24221de2b4a45c8c89482fff96ffb7e55

Algorithm 2 DeFi Token Swap Counting

Input: transaction set $\mathcal{T}$ of an incident

$\mathcal{T} \leftarrow \mathcal{T} - \{tx | tx.value == 0\}$;
$\mathcal{T} \leftarrow \mathcal{T} - \{tx | tx.from == tx.to\}$;
$\mathcal{T} \leftarrow \mathcal{T} - \{tx | tx.from == zeroaddress\}$;
$\mathcal{T} \leftarrow \mathcal{T} - \{tx | tx.to == zeroaddress\}$;
$numSwap \leftarrow 0$;
$i \leftarrow 0$, $j \leftarrow 1$;
for $i < |\mathcal{T}| - 1, j < |\mathcal{T}|$ **do**
  if $tx_i.hash == tx_j.hash$ & $tx_i.tokenSymbol \neq tx_j.tokenSymbol$ & $tx_i.from == tx_j.to$ & $tx_i.to == tx_j.from$ **then**
    tx_i and tx_j is a swap pair;
    $numSwap \leftarrow numSwap + 1$;
  end
  $i \leftarrow i + 1$, $j \leftarrow j + 1$;
end

that occurred in December 2021, the attackers quickly exchanged $5.7million worth of USDT stolen through the Curve.Fi service for DAI, USDC, in about two hours and 40 minutes.

- *Swapping tokens for mixing.* Many criminals make use of DEXs to swap their stolen tokens to ETH for mixing. For example, in the Bitmart Hack[16] event that occurred in December 2021, the criminal swapped MANA token for ETH in 1 inch DEX, then sent swapped ETH to Tornado.Cash for mixing.
- *Swapping tokens to bridge them to other blockchains.* Cross-chain transactions of criminals are cunning behavior to confuse the flow of dirty money. Before cross-chain transactions, criminals need to swap assets for tokens convertible on bridges. For example, in the Nexus Mutual Hacker[17] event that occurred in December 2020, the stolen ETH was swapped for renBTC, then the renBTC was bridged to the Bitcoin blockchain. RenBTC is a wrapped version of bitcoin on Ethereum which can then be bridged across to the Bitcoin blockchain using RenBridge.

b) Method II: counterfeit token deployment: In addition to ML techniques such as layered transfers and cross-chaining, more cunning hackers may disguise themselves as other common players to evade detection, where the existence of counterfeit tokens provides the perfect opportunity for hackers to launder money. Researchers [40], [41] have found that counterfeit tokens are prevalent in the blockchain ecosystem because most DEXs do not enforce any rules for token listing. Hackers can easily create counterfeit tokens and liquidity pools, and even disguise themselves as ordinary speculators to launder illicit funds from liquidity pools of counterfeit tokens. To this end, we will conduct an empirical analysis based on the counterfeit token dataset provided by Gao et al. [40], and

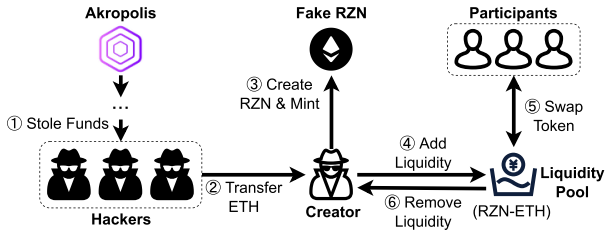


Fig. 11. The role fake tokens play in money laundering.

are surprised to find that in **13 of the 73** cases in this paper are related to counterfeit tokens some way.

One notable case is Akropolis Hacker[18] (a DeFi exploiter). By tracing the downstream transactions of Akropolis Hacker, we find evidence that this hacker was laundering money by creating counterfeit tokens. As plotted in Fig. 11, the main procedures are as follows:

1) Using the tracing method mentioned earlier, we see that the hacker cascaded the stolen funds from Akropolis to several accounts under his control ($0 \times 1c80$[19] and $0 \times 982d$[20] identified as "Heist" in our dataset), and transferred funds to another controlled account $0 \times 1f8$.[21]
2) Subsequently, address $0 \times 1f8$ created the **fake token RZN** (Rizen Token),[22] and a liquidity pool on Uniswap[23] with 594 fake RZN and 0.877 ETH.
3) The hackers then manipulated the liquidity pool through multiple accounts, posing as ordinary speculators and participating in the trading of counterfeit tokens, e.g. address $0 \times 8c4d$[24] sold 1000 fake RZN and got 300 clean ETH.
4) Finally, the RZN creator removed the liquidity of 2144 RZN coins and 0.25 ETH. The hackers successfully laundered the illegal funds by disguising their addresses as ignorant participants.

c) Method III: cheap sale of stolen NFTs: On Ethereum, besides fungible tokens resembling bank card assets (i.e., ERC20 tokens), there is a special type of token known as Non-Fungible Tokens (NFTs), often utilizing the ERC721 and ERC1155 token protocols. We observe that when these NFTs are stolen, criminals typically do not trade the stolen NFTs into other NFTs (similar to Method I). Instead, they tend to sell the stolen NFTs at a low price and convert them into assets of other protocols, such as ETH or USDT, which are fungible native tokens or ERC20 tokens.

For instance, on the day when Arthur's hot wallet was compromised, the hacker directly sold or auctioned the

[16] https://etherscan.io/address/0×39fb0dcd13945b835d47410ae0de7181d3edf270
[17] https://etherscan.io/address/0×09923e35f19687a524bbca7d42b92b6748534f25
[18] https://etherscan.io/address/0×9f26ae5cd245bfeeb5926d61497550f79d9c6c1c
[19] https://etherscan.io/address/0×1c80f8670f5c59aab8e81e954aabb64dabde2710
[20] https://etherscan.io/address/0 × 982dd33d6bc5 bf83eedcbcab92e4899c7a
[21] https://etherscan.io/address/0 × 1f84ba7bacd29e875367688b38ecccb7849b50fa
[22] https://etherscan.io/token/0 × 9c91310c9bf1c779b667f46322d33bfdc96c1a07
[23] https://etherscan.io/address/0×658b4a15aae288757c41a9b074ab1881d3ecad0c
[24] https://etherscan.io/address/0×8c4dedecbe3e8fbcc0501599cb59e7feadd99ffc

stolen 17 Azuki series NFTs at a price of around 10 ETH, significantly lower than the average market price at that time (13 ETH). The most significant price drop was observed in Azuki #606[25] — from 78 ETH ($105,506.70) before the heist to 50.15 WETH ($67,835.40). Subsequently, to avoid being frozen, the hacker transferred the stolen NFTs to another wallet before selling them. At this point, it becomes more challenging to differentiate whether the entity behind the address receiving the stolen NFTs is an address controlled by the hacker or a regular user, thereby increasing the difficulty of tracking the hacker's money laundering transactions. In terms of the impact on the NFT community, ordinary users purchasing stolen NFTs dumped by hackers at low prices could lead to a crisis of confidence in the NFT program. NFT trading platforms may flag stolen NFT assets long after the NFT heists have occurred, and ordinary users are likely to purchase these stolen NFTs accidentally, leading to them being subsequently blacklisted by NFT trading platforms. In July 2022, an OpenSea user posted a social media complaint that 88 days after he bought a CloneX NFT, the NFT was flagged as suspicious by the OpenSea platform, which is very unfair to ordinary buyers.

V. Discussion

From the perspective of practical applications, our framework can offer the following insights and assistance for regulatory agencies and security companies: (i) Supplement suspicious ML addresses and transactions. The scope of identification in our proposed framework extends from the source addresses in the placement set to the service providers. Security companies can leverage our AML tracking framework to trace suspicious money laundering accounts and transactions based on user-reported stolen addresses as placement addresses; (ii) Design risk signals based on observations and insights: A risk signal alert system can be devised based on insights on properties such as account transaction amounts, transaction frequencies, transaction network patterns, etc. Once one or more indicators reach a set threshold, it triggers a risk signal alert.

A. Future Work

In addition to the intricate ML transaction pathways, cryptocurrency AML systems also need to consider ML activities after the assets reach the destination service providers (such as cross-chain bridges or coin mixers). The pathways of money laundering beyond service providers, including coin mixing and cross-chain transactions, need to be further investigated in future work based on the specific types of service providers and their implementation mechanisms:

1) Coin Mixing Mapping: The research challenge of mixing mapping lies in the fact that mixers create random mapping relationships between several senders and receivers of transactions/accounts, making it difficult to establish explicit one-to-one mapping. Mapping coin mixing transactions requires the design of specific heuristic rules based on the paradigm of the mixing service (or the operational patterns adopted during mixing), as discussed in [42] and [43].

[25]https://etherscan.io/nft/0xed5af388653567af2f388e6224dc7c4b3241c544/606

2) Cross-Chain Transaction Tracking: The research challenge stems from the isolation of different blockchain spaces, leading to a lack of interoperability in ledger data between blockchains. During the process of initiating cross-chain transactions using a cross-chain bridge, there is no direct connection between the deposit transactions on the source chain and the withdrawal transactions on the destination chain. For future work, we plan to delve into transaction tracking for cross-chain bridges and intend to propose a DeFi cross-chain transaction tracking tool based on smart contract transactions, traces, and logs.

B. Ethical Considerations

In this paper, we reveal the first crypto ML dataset on Ethereum, investigating and analyzing the ML techniques of hackers, exploiters, scammers, and others. The disclosure and investigation may cause the community to worry about contributing to the "copycat crime" effect, but actually, our research motivation is similar to the studies of Ponzi contracts [44], phishing scams [45], DApp attacks [46], counterfeit tokens [40], etc. The money laundering accounts and transactions published in EthereumHeist are only the tip of the iceberg. As shown in this paper, cybercriminals are improving their methods and techniques year by year in the "cat-and-mouse" game of Ethereum AML, reinforcing the need for investigation and understanding of crypto ML. This work will facilitate more effective designs of AML algorithms based on our interesting findings in the laundering accounts and networks, and further promote the healthy development of the Ethereum ecosystem.

As for whether our research involves privacy issues, the answer is NO. First, the data we collect is completely public and can be accessed by anyone on the blockchain. Second, our dataset only includes anonymous transaction data on the blockchain, but not other data associated with real-life personal information. Therefore, based on these two points, we do not consider that this work will invade the privacy of others, or directly lead to the arrest or prosecution of individuals.

VI. Related Work

A. Cryptocurrency Anti-Money Laundering Techniques

In traditional financial scenarios, AML techniques rely on identity-linked information and diverse modeling and learning approaches to obtain and analyze money laundering data. However, in blockchain systems with pseudonymous accounts, the identity information and the association between accounts are usually not easily accessible. In the world of cryptocurrencies, the first publicly available dataset related to money laundering was the Elliptic dataset, classifying Bitcoin transactions into licit and illicit categories. The Elliptic dataset has attracted much attention and has been widely followed and used in a number of studies [14], [16], [47], [48]. The Elliptic dataset, however, is not suitable for developing and validating AML techniques on account-based blockchains (e.g., Ethereum) for two reasons. First, the dataset provides only binary labels for ML transactions and lacks further details on the events and stages of ML. Second, the Bitcoin

platform, which the dataset primarily focuses on, exhibits transaction behaviors that are significantly different from those of account-based blockchains due to its lack of support for smart contracts and decentralized applications. Therefore, for AML on account-based blockchains, a dataset that represents diverse transactions and behaviors is urgently needed to be proposed.

For Ethereum money laundering, there is some preliminary work based on the Upbit Hack incident. Fu et al. [49] discuss whether money laundering on Ethereum has traditional traits. Liu et al. [50] develop a graph embedding-based algorithm called GTN2vec, evaluating on the Upbit Hack dataset. However, most of the related literature focuses on the detection of illicit accounts [51], [52], [53], fraudulent transactions [54], [55], and abnormal smart contracts [56]. However, the above-mentioned papers mainly focus on the source of the stolen funds, but cannot uncover the subsequent money laundering activities.

B. Financial Security Issues on Blockchain

Security issues abound in the blockchain ecosystem, such as phishing scams, Ponzi schemes, wash trading and DApp attacks, etc. [37], [44], [45], [46], [57], [58]. There exist several datasets for anomaly detection and a series of approaches have been put forward to solve these issues. For example, Chen et al. [44] collect Ponzi schemes labels[26] and propose a Ponzi contract detection approach. Wu et al. [58] propose a network-embedding based method for phishing identification and disclose a phishing scam dataset.[27] Existing efforts are usually focused on the beginning of the security incidents without digging deeper into the money laundering behind them. It has been reported [8] that many security incidents are followed by money laundering to withdraw cash through service providers such as exchanges. As a result, existing research cannot fully understand the whole story of security incidents.

VII. Conclusion

In this paper, we present the first crypto-asset money laundering (ML) detection and analysis framework called XBlock-Flow, focusing specifically on the Ethereum blockchain platform. Inspired by observations on a realistic case, we propose a novel ML account identification method using Ethereum heist incidents as clues. In particular, starting from a very small number of security incident accounts labeled as heist, we extract a large number of ML transactions and related accounts, and construct a crypto-asset ML dataset named *EthereumHeist*. Compared with the existing Elliptic dataset which only provides binary labels for Bitcoin ML, *EthereumHeist* dataset contains much richer details, including information on the year, category, and amount stolen of the ML incidents; it also includes a three-stage division of the ML process that contains the layer information of the ML addresses, the detailed information of the ML transaction, and the category labels of the money laundering export. Based on this dataset, we conduct a comprehensive multi-perspective analysis for the three main stages of blockchain money laundering, i.e., placement, layering, and integration, and obtain a series of interesting findings, as well as observe some sophisticated money laundering methods like creating counterfeit tokens and masquerading as speculators. We believe that the proposed dataset and money laundering identification and analysis results shared in this paper can provide important insights for future work on blockchain money laundering behavior analysis and anti-money laundering technology design.

[26] https://www.kaggle.com/datasets/xblock/smart-ponzi-scheme-labels

[27] http://xblock.pro/#/dataset/6

References

[1] Chainalysis Team. (2022). *The Chainalysis State of Web3 Report*. [Online]. Available: https://go.chainalysis.com/2022-web3-report.html

[2] M. Swan, *Blockchain: Blueprint for a New Economy*. Sebastopol, CA, USA: O'Reilly Media, 2015.

[3] M. Xu, X. Chen, and G. Kou, "A systematic review of blockchain," *Financ. Innov.*, vol. 5, no. 1, pp. 1–14, 2019.

[4] M. Weber et al., "Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics," 2019, *arXiv:1908.02591*.

[5] J. Wu, J. Liu, Y. Zhao, and Z. Zheng, "Analysis of cryptocurrency transactions from a network perspective: An overview," *J. Netw. Comput. Appl.*, vol. 190, Sep. 2021, Art. no. 103139.

[6] Chainalysis Team. (2023). *The Chainalysis 2023 Crypto Crime Report*. [Online]. Available: https://go.chainalysis.com/2023-crypto-crime-report.html

[7] Chainalysis Team. (2021). *Report: Key Players of the Cryptocurrency Ecosystem*. [Online]. Available: https://go.chainalysis.com/rs/503-FAP-074/images/Key-players-in-crypto-report.pdf

[8] Chainalysis Team. (2022). *The Chainalysis 2022 Crypto Crime Report*. [Online]. Available: https://go.chainalysis.com/2022-crypto-crime-report.html

[9] Z. Gao and M. Ye, "A framework for data mining-based anti-money laundering research," *J. Money Laundering Control*, vol. 10, no. 2, pp. 170–179, May 2007.

[10] C. Wronka, "'Cyber-laundering': The change of money laundering in the digital age," *J. Money Laundering Control*, vol. 25, no. 2, pp. 330–344, 2021.

[11] B. Dumitrescu, A. Baltoiu, and S. Budulan, "Anomaly detection in graphs of bank transactions for anti money laundering applications," *IEEE Access*, vol. 10, pp. 47699–47714, 2022.

[12] J.-D.-J. Rocha-Salazar, M.-J. Segovia-Vargas, and M.-D.-M. Camacho-Miñano, "Money laundering and terrorism financing detection using neural networks and an abnormality indicator," *Exp. Syst. Appl.*, vol. 169, May 2021, Art. no. 114470.

[13] FATF. (2020). *Money Laundering and Terrorist Financing Red Flag Indicators Associated With Virtual Assets*. [Online]. Available: http://www.fatf-gafi.org/publications/fatfrecommendations/documents/Virtual-Assets-Red-Flag-Indicators.html

[14] I. Alarab, S. Prakoonwit, and M. I. Nacer, "Comparative analysis using supervised learning methods for anti-money laundering in Bitcoin," in *Proc. 5th Int. Conf. Mach. Learn. Technol.*, Jun. 2020, pp. 11–17.

[15] P. Xia, Z. Ni, H. Xiao, X. Zhu, and P. Peng, "A novel spatiotemporal prediction approach based on graph convolution neural networks and long short-term memory for money laundering fraud," *Arabian J. Sci. Eng.*, vol. 47, no. 2, pp. 1921–1937, Feb. 2022.

[16] J. Lorenz, M. I. Silva, D. Aparicio, J. T. Ascensao, and P. Bizarro, "Machine learning methods to detect money laundering in the Bitcoin blockchain in the presence of label scarcity," in *Proc. ICAIF*, 2020, pp. 1–8.

[17] Chainalysis Team. (2021). *Lazarus Group Pulled Off 2020's Biggest Exchange Hack and Appears to be Exploring New Money Laundering Options*. [Online]. Available: https://blog.chainalysis.com/reports/lazarus-group-kucoin-exchange-hack/

[18] S. M. Werner, D. Perez, L. Gudgeon, A. Klages-Mundt, D. Harz, and W. J. Knottenbelt, "SoK: Decentralized finance (DeFi)," 2021, *arXiv:2101.08778*.

[19] D. Lin. *Supplementary Material of 'Towards Understanding Crypto-Asset Money Laundering Through the Lenses of Ethereum Heists'*. Accessed: Nov. 8, 2023. [Online]. Available: https://github.com/lindan113/EthereumHeist

[20] Medium. *Drawing the Distinction Between the Uppercase 'b' and Lowercase 'b' in Bitcoin*. Accessed: Jul. 20, 2023. [Online]. Available: https://blockchain.medium.com/drawing-the-distinction-between-the-uppercase-b-and-lowercase-b-in-bitcoin-c37ae4464c22

[21] Investopedia. *Utxo Model: Definition, How it Works, and Goals*. Accessed: Jul. 20, 2023. [Online]. Available: https://www.investopedia.com/terms/u/utxo.asp

[22] V. Buterin, "A next-generation smart contract and decentralized application platform," Ethereum, White Paper 1, 2014. [Online]. Available: https://ethereum.org/669c9e2e2027310b6b3cdce6e1c52962/Ethereum_Whitepaper_-_Buterin_2014.pdf

[23] T. Hu et al., "Transaction-based classification and detection approach for Ethereum smart contract," *Inf. Process. Manag.*, vol. 58, no. 2, Mar. 2021, Art. no. 102462.

[24] T. Chen et al., "Understanding Ethereum via graph analysis," *ACM Trans. Internet Technol.*, vol. 20, no. 2, pp. 1–32, May 2020.

[25] Ethereum. *Erc-20 Token Standard*. Accessed: Jul. 20, 2023. [Online]. Available: https://ethereum.org/en/developers/docs/standards/tokens/erc-20/

[26] W. Chen, T. Zhang, Z. Chen, Z. Zheng, and Y. Lu, "Traveling the token world: A graph analysis of Ether eum ERC20 token ecosystem," in *Proc. Web Conf.*, Apr. 2020, pp. 1411–1421.

[27] Chainalysis Team. (2021). *Who's Who on the Blockchain? Mapping the Key Players in the Cryptocurrency Ecosystem*. Accessed: Jul. 20, 2023. [Online]. Available: https://go.chainalysis.com/mapping-key-players-in-crypto.html

[28] K. Qin, L. Zhou, Y. Afonin, L. Lazzaretti, and A. Gervais, "CeFi vs. DeFi—Comparing centralized to decentralized finance," 2021, *arXiv:2106.08157*.

[29] X. Li et al., "FlowScope: Spotting money laundering based on graphs," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 4731–4738.

[30] M. Möser, R. Böhme, and D. Breuker, "Towards risk scoring of Bitcoin transactions," in *Proc. FC*, 2014, pp. 16–32.

[31] X. T. Lee, A. Khan, S. Sen Gupta, Y. H. Ong, and X. Liu, "Measurements, analyses, and insights on the entire Ethereum blockchain network," in *Proc. Web Conf.*, Apr. 2020, pp. 155–166.

[32] D. Lin, J. Chen, J. Wu, and Z. Zheng, "Evolution of Ethereum transaction relationships: Toward understanding global driving factors from microscopic patterns," *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 2, pp. 559–570, Apr. 2022.

[33] Z. Wu, J. Liu, J. Wu, Z. Zheng, and T. Chen, "TRacer: Scalable graph-based transaction tracing for account-based blockchain trading systems," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 2609–2621, 2023.

[34] Z. Wu, J. Liu, J. Wu, Z. Zheng, X. Luo, and T. Chen, "Know your transactions: Real-time and generic transaction semantic representation on blockchain & web3 ecosystem," in *Proc. ACM Web Conf.*, Apr. 2023, pp. 1918–1927.

[35] P. Zheng, Z. Zheng, J. Wu, and H.-N. Dai, "XBlock-ETH: Extracting and exploring blockchain data from Ethereum," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 95–106, 2020.

[36] A. R. Benson, D. F. Gleich, and J. Leskovec, "Higher-order organization of complex networks," *Science*, vol. 353, no. 6295, pp. 163–166, Jul. 2016.

[37] F. Victor and A. M. Weintraud, "Detecting and quantifying wash trading on decentralized cryptocurrency exchanges," in *Proc. Web Conf.*, Apr. 2021, pp. 23–32.

[38] FATF. (2019). *Guidance for a Risk-Based Approach to Virtual Assets and Virtual Asset Service Providers*. [Online]. Available: www.fatf-gafi.org/publications/fatfrecommendations/documents/Guidance-RBA-virtual-assets.html

[39] FATF. (2021). *Updated Guidance for a Risk-Based Approach to Virtual Assets and Virtual Asset Service Providers*. [Online]. Available: www.fatf-gafi.org/publications/fatfrecommendations/documents/Updated-Guidance-RBA-VA-VASP.html

[40] B. Gao et al., "Tracking counterfeit cryptocurrency end-to-end," in *Proc. ACM SIGMETRICS/Int. Conf. Meas. Model. Comput. Syst.*, May 2021, pp. 1–28.

[41] P. Xia et al., "Trade or trick? Detecting and characterizing scam tokens on uniswap decentralized exchange," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 5, no. 3, pp. 1–26, 2021.

[42] L. Wu et al., "Towards understanding and demystifying Bitcoin mixing services," in *Proc. Web Conf.*, Apr. 2021, pp. 33–44.

[43] Z. Wang et al., "On how zero-knowledge proof blockchain mixers improve, and worsen user privacy," in *Proc. ACM Web Conf.*, Apr. 2023, pp. 2022–2032.

[44] W. Chen, Z. Zheng, J. Cui, E. Ngai, P. Zheng, and Y. Zhou, "Detecting Ponzi schemes on Ethereum: Towards healthier blockchain technology," in *Proc. World Wide Web Conf.*, 2018, pp. 1409–1418.

[45] S. Li, G. Gou, C. Liu, C. Hou, Z. Li, and G. Xiong, "TTAGN: Temporal transaction aggregation graph network for Ethereum phishing scams detection," in *Proc. ACM Web Conf.*, Apr. 2022, pp. 661–669.

[46] L. Su et al., "Evil under the sun: Understanding and discovering attacks on Ethereum decentralized applications," in *Proc. USENIX Secur.*, 2021, pp. 1307–1324.

[47] K. Kolachala, E. Simsek, M. Ababneh, and R. Vishwanathan, "SoK: Money laundering in cryptocurrencies," in *Proc. 16th Int. Conf. Availability, Rel. Secur.*, Aug. 2021, pp. 1–10.

[48] A. B. Turner, S. McCombie, and A. J. Uhlmann, "Discerning payment patterns in Bitcoin from ransomware attacks," *J. Money Laundering Control*, vol. 23, no. 3, pp. 545–589, Jul. 2020.

[49] Q. Fu, D. Lint, Y. Cao, and J. Wu, "Does money laundering on Ethereum have traditional traits?" in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2023, pp. 1–5.

[50] J. Liu et al., "Graph embedding-based money laundering detection for Ethereum," *Electronics*, vol. 12, no. 14, p. 3180, Jul. 2023.

[51] S. Farrugia, J. Ellul, and G. Azzopardi, "Detection of illicit accounts over the Ethereum blockchain," *Exp. Syst. Appl.*, vol. 150, Jul. 2020, Art. no. 113318.

[52] F. Poursafaei, G. B. Hamad, and Z. Zilic, "Detecting malicious Ethereum entities via application of machine learning classification," in *Proc. 2nd Conf. Blockchain Res. Appl. Innov. Netw. Services (BRAINS)*, Sep. 2020, pp. 120–127.

[53] D. Lin, J. Wu, T. Huang, K. Lin, and Z. Zheng, "Who is who on Ethereum? Account labeling using heterophilic graph convolutional network," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Nov. 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10324318, doi: 10.1109/TSMC.2023.3329520.

[54] R. M. Aziz, M. F. Baluch, S. Patel, and A. H. Ganie, "LGBM: A machine learning approach for Ethereum fraud detection," *Int. J. Inf. Technol.*, vol. 14, no. 7, pp. 3321–3331, Dec. 2022.

[55] Q. Umer, J.-W. Li, M. R. Ashraf, R. N. Bashir, and H. Ghous, "Ensemble deep learning-based prediction of fraudulent cryptocurrency transactions," *IEEE Access*, vol. 11, pp. 95213–95224, 2023.

[56] L. Liu, W.-T. Tsai, M. Z. A. Bhuiyan, H. Peng, and M. Liu, "Blockchain-enabled fraud discovery through abnormal smart contract detection on Ethereum," *Future Gener. Comput. Syst.*, vol. 128, pp. 158–166, Mar. 2022.

[57] W. Chen, J. Wu, Z. Zheng, C. Chen, and Y. Zhou, "Market manipulation of Bitcoin: Evidence from mining the Mt. Gox transaction network," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2019, pp. 964–972.

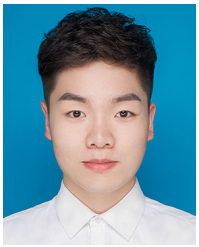
[58] J. Wu et al., "Who are the phishers? Phishing scam detection on Ethereum via network embedding," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 2, pp. 1156–1166, Feb. 2022.

Jiajing Wu (Senior Member, IEEE) received the B.Eng. degree in communication engineering from Beijing Jiaotong University, Beijing, China, in 2010, and the Ph.D. degree from The Hong Kong Polytechnic University, Hong Kong, in 2014.

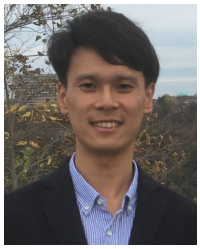
She is currently an Associate Professor with the School of Software Engineering, Sun Yat-sen University, Zhuhai, China. Her research interests include blockchain, graph mining, and network science. She was awarded the Hong Kong Ph.D. Fellowship Scheme during her Ph.D. study in Hong Kong (2010–2014).

Dan Lin (Graduate Student Member, IEEE) received the B.Eng. degree in software engineering from Sun Yat-sen University, Guangzhou, China, in 2019, where she is currently pursuing the Ph.D. degree with the School of Software Engineering. Her current research interests include blockchain, cryptocurrency, theories and applications of network science, and anti-money laundering.

Qishuang Fu received the B.Eng. degree in automation from the Dalian University of Technology, Dalian, China, in 2021. She is currently pursuing the M.Sc. degree with the School of Computer Science and Engineering, Sun Yat-sen University. Her current research interests include blockchain, risk management, and anti-money laundering.



Shuo Yang (Graduate Student Member, IEEE) received the B.Eng.Mgt. degree in management information system from the Zhongnan University of Economics and Law, Wuhan, China, in 2022. He is currently pursuing the Ph.D. degree with the School of Software Engineering, Sun Yat-sen University, Zhuhai, China. His research interests include NFT, Web3, smart contract security, and software analysis.



Ting Chen (Member, IEEE) received the B.S. and Ph.D. degrees from the University of Electronic Science and Technology of China, Chengdu, Sichuan, China, in 2007 and 2013, respectively.

He is currently a Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interests include blockchain security, software engineering, cyberphysical industrial systems, and application of network security.

Zibin Zheng (Fellow, IEEE) is currently a Professor and the Deputy Dean of the School of Software Engineering, Sun Yat-sen University, Zhuhai, China. He authored or coauthored more than 200 international journal and conference papers, including one ESI hot paper and ten ESI highly cited papers. According to Google Scholar, his papers have more than 28,000 citations. His research interests include blockchain, software engineering, and services computing.

Bowen Song received the Ph.D. degree in applied math and statistics from Stony Brook University in 2015. He joined Ant Group in 2017, where he is currently a Senior Staff Algorithm Engineer with the Anti-Money-Laundering Algorithm Team. His research interests include behavior sequential learning, deep graph learning, and their applications in financial risk management and web3.