Beyond a Fistful of Tumblers: Toward a Taxonomy of Ethereum-based Mixers

Completed Research Paper

Tom Barbereau, Egor Ermolaev,
Martin Brennecke, Eduard Hartwich, Johannes Sedlmeir
SnT, University of Luxembourg
Luxembourg, Luxembourg
{tom.barbereau, egor.ermolaev,
martin.brennecke, eduard.hartwich, johannes.sedlmeir}@uni.lu

Abstract

The role played by decentralised services in the obfuscation of crypto-asset transactions performed on transparent blockchains has increasingly captured the attention of regulators. This is exemplified by the headlines about the U.S. Treasury's sanctions on the Ethereum-based mixer Tornado Cash. Yet, despite the existing controversies on the use of mixers, the different functionalities of these information systems with an inherent dark side remain to be explored by the literature. So far, contributions primarily encompass technical works and studies that focus on the Bitcoin ecosystem. This paper puts forward a multi-layer taxonomy of the smart-contract-based – and, therefore, functionally richer – family of mixers on Ethereum. Our proposed taxonomy is grounded on (1) a review of existing literature, (2) an analysis of mixers' project documentation, (3) their corresponding smart contracts, and (4) expert interviews. Our evaluation included the application of the taxonomy to two mixers – RAILGUN and zkBob. The taxonomy represents a valuable tool for law enforcement, regulators, and other stakeholders to explore critical properties affecting compliance and use of Ethereum-based mixers.

Keywords: Anonymisation, classification, compliance, crypto-asset, digital forensics, zero-knowledge proof

Introduction

Despite the transparent nature of public blockchains and the possibility to de-anonymise pseudonymous addresses by linking transactions (Biryukov et al., 2014; Meiklejohn et al., 2013), the extent of criminal activities in the crypto-asset space reached an all-time high in 2022 (Chainalysis, 2023). One of the key tools facilitating these activities are information systems that can be used to obfuscate transaction graphs: so-called crypto-asset "tumblers" or "mixers". The total value of crypto-assets processed by these systems in 2022 alone is estimated to stand at around $10 billion, with at least 24 % originating from illicit sources (Chainalysis, 2023). In August 2022, the public saw the tip of the criminal iceberg when the U.S. Department of the Treasury (2022) sanctioned the Ethereum-based mixer Tornado Cash for "laundering" some $7 billion in crypto-assets; of which $455 million were linked to the cyber-criminal Lazarus Group affiliated with the North Korean regime. In an unprecedented turn, the Treasury's Office of Foreign Assets Control (OFAC) listed smart contract addresses associated with Tornado Cash – making it illegal for U.S. citizens to receive or send assets through this service (Nadler & Schär, 2023). What followed is a cascade of events: the Dutch

authorities arrested one of the mixer's co-founders (FIOD, 2022), the source code on GitHub was deleted and later republished (Shen, 2022), and most recently, the other two co-founders of Tornado Cash were arrested (U.S. Department of the Treasury, 2023).

Typically, scholars differentiate between mixers that are (A) provided by a trusted third-partys (TTPs) acting as custodians; (B) implemented in non-custodial wallets, based on peer-to-peer interactions with other users; or (C) run in a non-custodial and non-interactive way, based on smart contracts that allow users to make the deposit and withdrawal of assets unlinkable with the help of sophisticated cryptographic tools (Béres et al., 2021; Nadler & Schär, 2023; Trozze et al., 2023). Type C is exclusive to smart contract blockchains like Ethereum and arguably most popular given the superior obfuscation provided by zero-knowledge proofs (ZKPs) (Burleson et al., 2022; Nadler & Schär, 2023; Wang et al., 2022).Thus far, mixers' functionalities remain largely understudied beyond research in computer science. Related work has focused on technical improvements to performance (Cirkovic et al., 2022), opportunities to improve anonymity guarantees (Wang et al., 2022), and how to design accountability mechanisms and compliance features (Burleson et al., 2022). To our knowledge, the most comprehensive secondary studies on mixers are those of Feng et al. (2019) and Pakki et al. (2021). Both are classifications that primarily focus on Bitcoin-based mixers. This is a considerable limitation for two reasons: (1) in parallel to an increase in illicit activities on the Ethereum blockchain and superior anonymisation techniques, the corresponding Ethereum-based mixers became those most used (Wang et al., 2022), and (2) the flexibility of smart contracts allows for greater variability in functionalities that impact, for instance, governance-related, economic, and regulatory aspects. We found only two works that address these type C mixers, and both focus solely on Tornado Cash. Nadler and Schär (2023) analysed its design and transaction history. Béres et al. (2021) attempted to de-anonymise transactions processed with it. Hence, to distinguish different aspects related to the functional level of mixers, we pose the following research question: *How to classify characteristics of non-custodial, non-interactive, Ethereum-based mixers?*

This paper systematically derives a taxonomy of Ethereum-based mixers following Nickerson et al. (2013) and Kundisch et al. (2021). Taxonomies are considered a suitable artefact to classify, categorise, and systematise objects and clarify complex fields of interest (Bailey, 1994). Our scope is limited to Ethereum-based mixers implemented via smart contracts. These platforms utilise the Ethereum Virtual Machine (EVM) or EVM-compatible virtual machines (Wang et al., 2022). Our taxonomy development comprises four cycles that respectively consider (1) existing peer-reviewed works, (2) the public documentation of four Ethereum-based mixers, (3) their underlying smart contracts, and (4) expert interviews. Following Kundisch et al. (2021), we further evaluate the final taxonomy by applying it to two mixers, RAILGUN and zkBob.

Our taxonomy contributes to the IS literature on "digital forensics" (Nance et al., 2009) and the "dark side of information technology use" (D'Arcy et al., 2014; Tarafdar et al., 2013). Its development process is inspired by a body of research that combines both on-chain and off-chain data sources related to blockchain-based systems (e.g., Barbereau et al., 2023; Zheng et al., 2021). Our research also responds to calls by the Basel Institute on Governance and Europol (2022) for a greater "understanding of crypto-assets and [associated] services", given that "money laundering schemes can often involve multiple blockchains and mixers located in different jurisdictions". Our research is of a descriptive, explanatory nature (Gregor, 2006): it allows practitioners to understand functionalities and distinguish between mixers more accurately. Classification, including taxonomies, can represent valuable tools as part of the evidence collection process and intelligence cycle in law enforcement and criminal intelligence (see, e.g., Fröwis et al., 2020; Sarre et al., 2018).

Background

In this section, we introduce the technical foundations of Ethereum-based mixers, including a description of the properties and use of ZKPs in this context. Its aim is less to provide a detailed description of each potential design choice of an implementation, but rather to provide a high-level perspective of one foundational and prevalent approach to implementing smart-contract-based mixers. Second, the motivation to provide these foundations lies in the prevalent use of jargon and concepts from computer science and cryptography. By introducing these accordingly, we aim to make our paper accessible to readers that perhaps lack a deep technical background. We restrict our scope to concepts indispensable for the reader to grasp

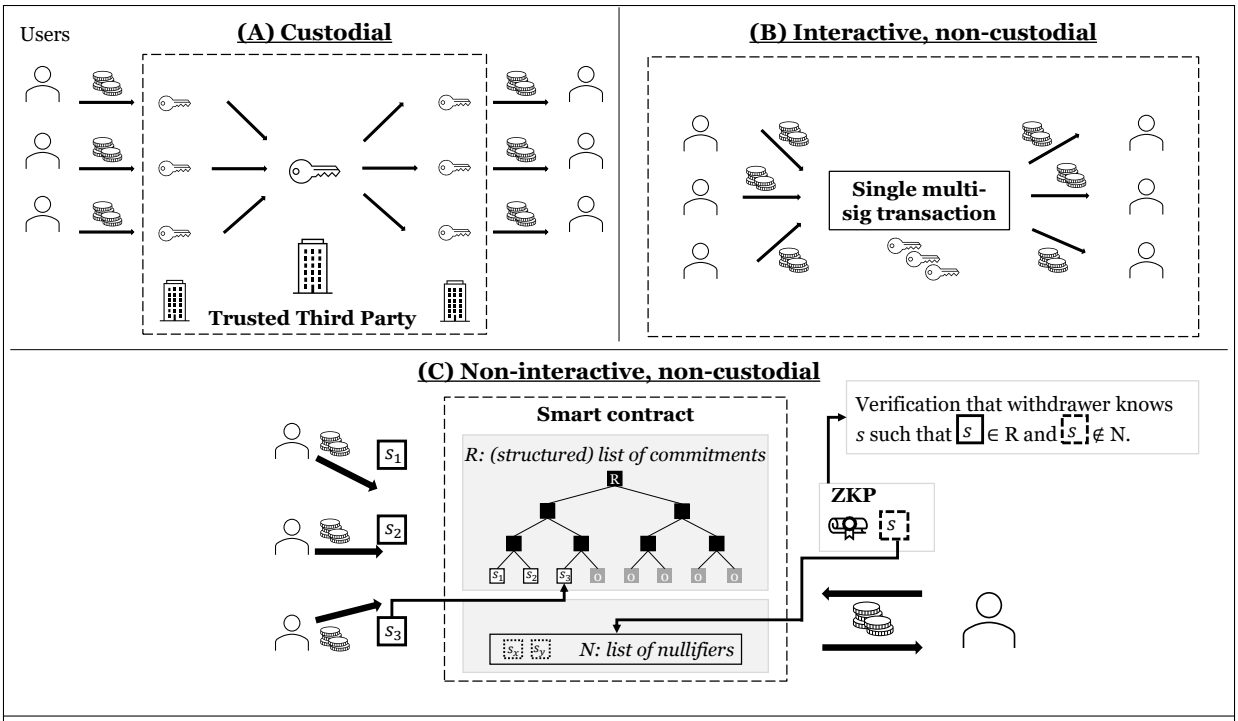

Figure 1. High-Level Perspective on Mixer Implementations

the different dimensions and characteristics later featured as part of the developed taxonomy. Third, establishing these allows for greater generalisability for consideration of projects beyond Ethereum, including other blockchains and decentralised applications that include payment functionalities and navigate privacy-related and regulatory requirements.

Like wallets, mixers can be implemented based on custodial (centralised) and non-custodial (decentralised) designs (Barbereau & Bodó, 2023; Béres et al., 2021). Figure 1 illustrates how mixers can be implemented from a high-level perspective. Custodial mixers (**A**) involve a TTP that offers a service such that entities can send their crypto-assets to an address associated with this TTP as a deposit. The TTP then applies a sequence of joining and splitting transactions with crypto-assets received from other addresses until a satisfactory degree of mixing is achieved. Finally, the TTP transfers the assets back to new addresses as specified by the depositors in bilateral communication (Nadler & Schär, 2023). Users can also engage in interactions to perform non-custodial mixing (**B**), for instance, by engaging in bilateral atomic swaps or CoinJoin-based approaches (Ghesmati et al., 2022). In the latter, each involved user specifies a sending and receiving address and communicates them multi-laterally to participants in the mixing process. The participants then create and jointly authorise (through every participant's digital signature with the cryptographic key corresponding to the sending address) a single transaction that consumes crypto-assets from all sending addresses and sends them back to the specified receiving addresses (Ruffing et al., 2014).

The trust required in a TTP that takes the custody of funds and learns about the connection between deposits and withdrawals (Nadler & Schär, 2023), as well as the complexity and data leakage risk associated with coordinating a substantial number of wallets in interactive approaches, make non-custodial, non-interactive mixers (**C**) a popular alternative. Here, users interact with a smart contract that provides the mixing service directly. This type typically uses non-interactive ZKPs to obfuscate transaction graphs. Deposited crypto-assets are mixed by avoiding the linkability between deposits and withdrawals. ZKPs represent a cryptographic primitive that allows a "prover" to convince a group of "verifiers" (and, therefore, in particular a blockchain-based smart contract) with a single message that a certain computation that the prover performed is correct, without requiring re-execution on the verifier side and in particular the disclosure of inputs, intermediary results, or outputs unless explicitly specified (Goldreich & Oren, 1994).

Non-custodial, non-interactive mixers typically store a structured, append-only list R ("commitments", often in the form of a Merkle tree) and a not necessarily structured, append-only list of nullifiers N. Elements of R and N are computed by users with two different hash functions and then sent to the mixing smart contract. Hash functions represent cryptographic one-way functions that map any input to a fixed-length output (e.g., 256 bits) and that do not allow deriving properties of the input from the output (Nadler & Schär, 2023). When a user wants to deposit crypto-assets, s/he creates a random secret (also called a *note*) s, hashes it with the hash function h_1, and sends crypto-assets and $h_1(s)$ to the mixing smart contract. The mixing smart contract increases its balance ('shielded pool of crypto-assets') by the sent amount and appends $h_1(s)$ to R. When a user seeks to withdraw crypto-assets that s/he previously deposited, s/he can read the current state of R and create a ZKP that they know some secret s such that $h_1(s) \in R$. Notably, neither $h_1(s)$ nor its position in R are revealed in this process. To prevent double-withdrawal despite unlinkability, the user also sends the hash of the secret with another hash function $h_2(s)$ and a ZKP that $h_2(s)$ was correctly computed from the same s underlying $h_1(s)$. The smart contract then appends $h_2(s)$ to the list of nullifiers N and verifies that the value $h_2(s)$ has not been added to it before. The ZKP avoids the creation of an observable link between depositing and withdrawal transaction, as for any observer, all commitments in R are equally likely to correspond to the withdrawal when ignoring potential time-related patterns (Wang et al., 2022).

Related work

Despite cryptocurrencies' decentralised nature and the use of public keys or hashes thereof as pseudonyms, it is possible to de-anonymise users leveraging the public visibility of transaction graphs. Meiklejohn et al. (2013)'s paper *A fistful of Bitcoins* presents one of the first methods to trace and identify crypto-asset users. It was eventually used by the FBI to build a case against the infamous Silk Road. In the following years, other works refined this approach or explored alternative ways to de-anonymise transactions on different blockchains; for instance, by incorporating Bitcoin's network topology (Biryukov et al., 2014) and network traffic (Biryukov & Tikhomirov, 2019) in their analyses. As Bitcoin's unspent transaction output (UTXO) model offers better privacy guarantees than account-based models underlying Ethereum and many other smart contract blockchains, Bitcoin has originally attracted more illicit activities (see also, Möser et al., 2013) and more research on de-anonymisation (Pocher et al., 2023). However, researchers also focused on transactions that went through more sophisticated Ethereum-based mixers like Tornado Cash (Béres et al., 2021) or cryptocurrencies with similar privacy features like Zcash (Biryukov & Tikhomirov, 2019).

Mixers can be seen as a technology whose use is fundamentally dual. On the one hand, these sophisticated anonymisation techniques on public blockchains can be used to address legitimate privacy concerns originating from the replicated transaction processing and storage on blockchains (Sedlmeir et al., 2022), to ensure fungibility (Biryukov et al., 2019), or to uphold (financial) privacy rights, for instance, in countries that suppress political opposition by foreclosing access to financial services (Campbell-Verduyn, 2018). On the other hand, because of their ability to provide a high degree of anonymity, mixers make it difficult to hold their users accountable for activities prior to use. As such, they enable malicious actors to escape regulatory efforts and bypass sanctions in the banking system. Academic research in digital investigations, criminology, and law observed that mixers are frequently used to do so (Trozze et al., 2023). This observation is supported by publications of (inter-)governmental law enforcement agencies (Europol, 2022; Interpol, 2020).

IS research has a standing history of analysing both the potential opportunities given by technological advances and critically reflecting on associated threats. Scholars of dedicated communities focus on topics of "digital forensics" (Nance et al., 2009) and the "dark side of information technology use" (D'Arcy et al., 2014; Tarafdar et al., 2013). In the latter, objects of study included but were not limited to social media (Chan et al., 2019), healthcare systems (Califf et al., 2020), and artificial intelligence (Mikalef et al., 2022). Although the IS literature on mixers is scarce, some works of its communities did consider crypto-assets more generally. For instance, Dhillon (2016) studied how novel technologies and, in particular, cryptocurrencies enable and facilitate money laundering.

Our research into how Ethereum-based crypto-asset mixers may be classified deviates from related work in three fundamental ways. First, several existing studies centre around mixer usage, with some of them focusing primarily on an assessment of the respective mixer's anonymity guarantees (Béres et al., 2021; Nadler & Schär, 2023). Second, previous publications contributed to our understanding of mixers by attempts to de-

anonymise transactions on crypto-asset mixers (Béres et al., 2021; See, 2023). Third, some studies analyse different Bitcoin-based mixing services with limited functionality (Ghesmati et al., 2022; Pakki et al., 2021) or focus on just a singular case, primarily Tornado Cash (Nadler & Schär, 2023). Previous studies thus do not provide an overview of the rich design options surrounding Ethereum-based mixers. We contribute to the latter discourse, as our taxonomy demonstrates that the design space for these mixers is much richer than previously studied, individual examples suggest. A comprehensive understanding of these design options may benefit future studies analysing the (criminal) use and compliance of Ethereum-based mixers.

Research Approach

To lift the fog over non-custodial, non-interactive, Ethereum-based crypto-asset mixers, we develop a taxonomy. Taxonomy building is suitable to classify objects with common characteristics, which, in turn, allows to organise subject areas and knowledge systematically (Bailey, 1994). Since taxonomies can serve as a foundation for future considerations in research and practice, our contribution addresses IS scholars, regulators, law enforcement, and digital forensics professionals. The development of multi-layer taxonomies is common for research on blockchain and crypto-assets given their still nascent status and rapid evolution (see, e.g., Hartwich et al., 2022; Ziegler & Welpe, 2022). Informed and inspired by these works, we follow the taxonomy-building process proposed by Nickerson et al. (2013) and later extended by Kundisch et al. (2021). While Nickerson et al. (2013) define seven steps, we also see relevance in the evaluation proposed by Kundisch et al. (2021). Figure 2 features our 10-step taxonomy development process. We opted for developing a multi-layer taxonomy. The introduction of layers unlocks the potential of including aggregation levels, which aid with structuring and classifying the dimensions and characteristics of the underlying object. We derived these layers inductively from our collected and analysed data and deductively from classifications in previous works.

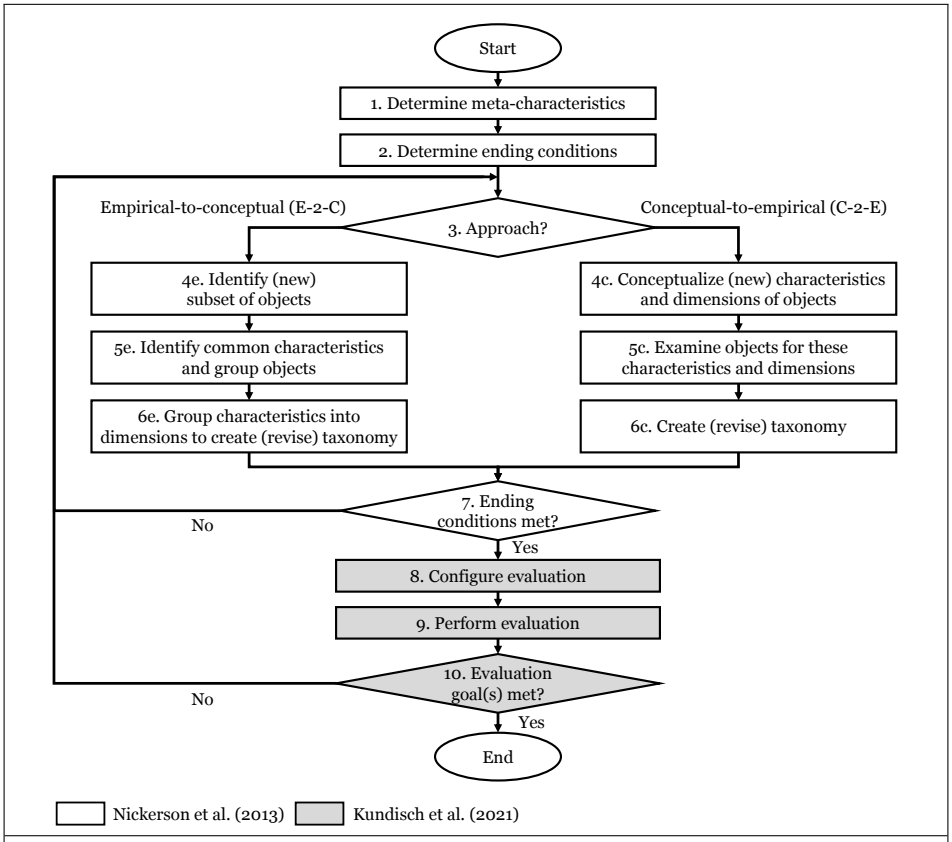

Figure 2. Taxonomy Development Steps

Definition of objectives (1–2): To support our target group of researchers, law enforcement, and regulators in classification and differentiation efforts, we define 'characteristics and functionalities of Ethereum-based mixers' as our meta-characteristic. We draw on the objective and subjective ending conditions proposed by Nickerson et al. (2013). For objective conditions, we summarise the ones suggested by Nickerson et al. (2013) and define that (1) a meaningful sample of mixers is analysed, (2) no layers (Ls), dimensions (Ds), or characteristics (Cs) were added, merged, or split, and (3) all dimensions provide "collectively exhaustive characteristics" (Nickerson et al., 2013, p.8). We also define subjective ending conditions, demanding that our taxonomy be concise, comprehensive, extendable, explanatory, and useful. As such, we check our taxonomy for all conditions defined by Nickerson et al. (2013) and Kundisch et al. (2021).

Design and development (3–7): Given that little information and data on Ethereum-based crypto-asset mixers is available, we follow a mixed approach consisting of four iterations (see Table 1). We started with a conceptual-to-empirical (C-2-E) approach for the first two iterations. We based this first iteration of taxonomy development on the most relevant literature identified via a search of the Scopus and IEEE Xplore databases, using "(mixer OR tumbler) AND (crypto OR Ethereum OR Bitcoin)" as a search string. Our search yielded 48 results on Scopus and 23 on IEEE Xplore. Out of these, we included only publications that are peer-reviewed and available in English. After title screening, abstract screening, and removing duplicates, five relevant publications remained. We hence decided to expand our literature base with two preprints by researchers that had previously published on mixers. For the first draft of our taxonomy, we built particularly on Feng et al. (2019) and Pakki et al. (2021) and supplemented valuable information from Amiram et al. (2022), Barbereau and Bodó (2023), Béres et al. (2021), Nadler and Schär (2023), Rathore et al. (2022), and van Wegberg et al. (2018). Informed by this literature base, we then considered four cases' project documentations (Table 2). Given that no guarantee was provided that project documentation in our specific cases matched the deployed source code of the respective mixer, we classified project documentation under the C-2-E cycle – even though it might also be considered empirical evidence in an empirical-to-conceptual (E-2-C) cycle. For the third iteration, we employed an E-2-C approach which we based on empirical data in the form of the smart contracts corresponding to the mixers in Table 2. Using smart contracts as an empirical data source is well-suited to meet our objectives. It complements project documentation on mixers, which we often found to be incomplete regarding key technical design choices or even differing from the coded reality. Since our goal is to analyse the characteristics and functionalities of crypto-asset mixers, it is paramount to screen the deployed smart contracts. We analysed the smart contracts written in Solidity and dissected their functionalities by mapping code snippets to the taxonomy's layers, dimensions, and characteristics. Table 3 features four examples of this coding scheme for fragments of mixers' Solidity code. For instance, in code fragment 1, the appearance of `_coinDenomination` in the constructor shows that the mixer defines the denomination at the moment of the contract's deployment. Thereby it corresponds to the layer of ***Mixer functionalities*** and the **Denomination** dimension, indicating that the characteristic is *Pre-defined*.

For the fourth iteration, we conducted expert interviews with seven researchers. Interviews are well suited for conducting exploratory and complex studies (Schultze & Avital, 2011). Each interview lasted between

It.	App.	Data Source	L/D/C	Status	Ending conditions
1	C-2-E	Literature review	6/15/47	Defined initial set of Ds and Cs.	Subj. and obj. ending conditions not met: Taxonomy is not explanatory without subject knowledge.
2	C-2-E	Project doc.	7/21/67	Added 6 Ds and 20 Cs. Clustered Ds into 7 Ls.	Obj. ending conditions not met: Cs were added and split, requiring another iteration of the process.
3	E-2-C	Smart contracts	5/20/47	Merged Ls to a total of 5, removed 1 D, merged or removed Cs.	Obj. ending conditions not met: Cs were deleted and added, requiring another iteration of the process.
4	E-2-C	Expert interviews	5/16/45	Removed or merged 4 Ds and 2 Cs. Adapted Ds and Cs.	All ending conditions met.

Table 1. Taxonomy Development Iterations

Case	Documentation	Ethereum smart contract address
Tornado Cash Nova	Tornado Cash Team (2021)	0xca0840578f57fe71599d29375e16783424023357
Cyclone	Cyclone Community (2023)	0xd619c8da0a58b63be7fa69b4cc648916fe95fa1b
Typhoon Cash	TyphoonCash (2021)	0x9cdb933edab885bb767658b9ed5c3800bc1d761b
Aztec	Andrews (2020)	0x737901bea3eeb88459df9ef1be8ff3ae1b42a2ba

Table 2. Cases Considered in Taxonomy Development

30 and 60 minutes (40 minutes average). We conducted the interviews between February and April 2023. To ensure rigour, we recorded and transcribed the interviews for further systematic analysis. We used expert sampling to select interviewees from academia based on recent publications. Although we contacted practitioners from three leading blockchain data analytics firms, they did not respond to our inquiries. Considering recent regulatory measures and increased scrutiny, we decided not to contact developers and users of non-custodial mixers, who may opt to remain anonymous. In each interview, we went through the layers in a semi-structured manner (Myers & Newman, 2007). We asked for the relevance of each dimension in the layer and whether the characteristics are appropriate and exhaustive. After the fourth interview, despite noticing saturation, we conducted three more interviews to ensure that no further layer, dimension, or characteristic had to be added or edited.

Evaluation (8–10): To evaluate our taxonomy, we also used these later interviews to gather feedback on its usefulness. To substantiate the objective validity of the taxonomy, we conducted an additional ex-post evaluation (Kundisch et al., 2021) in which we applied our taxonomy to the mixers RAILGUN and zkBob. This exercise verified that we met our ending conditions and concluded the taxonomy-building process.

In summary, and as indicated in Table 1, the taxonomy underwent various changes over four iterations. The first iteration established initial categories based on the literature review and introduced an "uncategorized" section for miscellaneous features. In the second iteration, the taxonomy shifted towards a more structured approach, while introducing more detailed dimensions. The third iteration refined this structure based on insights from mixers' smart contracts and rearranged certain dimensions. Finally, the fourth iteration further refined terminology based on feedback and integrated and removed various features due to its lack of justification from the mixer code-base.

Taxonomy of Ethereum-based Crypto-asset Mixers

Figure 3 displays the final taxonomy of non-custodial, non-interactive, Ethereum-based mixers. The layers of the taxonomy correspond to smart contracts' lifecycle (Sánchez-Gómez et al., 2020) and the methods they expose for the two core interactions users engage in. ***General features*** describes the choices developers make when designing the smart contract structure and characteristics of the environment in which the mixing service is deployed. The ***Terms of use*** layer describes the prerequisites that users need to fulfil to interact with the mixer (e.g., in terms of fees and compliance). They are reflected by the Solidity modifiers

№	Mixer / Address	Code snippet	Layer / Dimension	Char.
1	Cyclone	`uint256 public coinDenomination;` `constructor(…, uint256 _coinDenomination, …) {…}`	Mixer Func. / Denomination	Pre-defined
2	Cyclone	`function deposit(…) external payable nonReentrant {…` `  uint256 refund = msg.value - coinDenomination; …}`	Deposit Func. / Restitution	Refund
3	Cyclone	`address public govDAO;` `modifier onlyGovDAO { require(msg.sender == govDAO, "..."); _; }`	General Feat. / Governance	DAO

4	Typhoon Cash	`function _processDeposit() internal {`
		`  require(msg.value == 0, "...");`
		`  _safeErc20TransferFrom(…, denomination); }`
			Mixer Func. / Crypto-asset type	ERC20

Table 3. Exemplary Solidity Snippets and Mapped Characteristics

which restrict access to state updates (write methods) exposed by mixers. The remaining layers represent the user flow. We distinguish methods in the mixer smart contract that are uniquely connected to either ***Deposit functionalities*** or ***Withdrawal functionalities***. We associate dimensions that can equally be associated with both deposit and withdrawal functionalities to ***Mixer functionalities***. For instance, the crypto-asset type affects both deposit and withdrawal functionalities because it defines whether Ethereum's native token functionality or an external ERC20 interface will be used within the mixer to handle and transition the assets. In the following, we introduce each layer, dimension, and characteristic of the taxonomy and discuss their meaning, relevance, and implications. We also indicate whether characteristics within a dimension are mutually exclusive (ME), i.e., any mixer is characterised by a unique characteristic in this dimension (Y), or multiple characteristics may apply (N).

General features

Deployment layer refers to the environment in which the mixer is deployed and transactions are executed. *Layer 1 (L1)* refers to the native execution layer of Ethereum, which provides the infrastructure for smart contract-based transactions and the storage of corresponding states. Besides running the EVM, L1 controls the consensus protocol. Layer 2 (L2) refers to solutions that improve scalability and reduce transaction fees. L2 solutions are deployed on top of smart contracts and reduce the complexity of transaction processing on L1. A common family of L2 protocols is referred to as rollups. The two archetypes are optimistic and validity (often called zk-) rollups (Buterin, 2021). The former are based on the assumption that transactions are normally legitimate and perform smart-contract-based verifications of their validity only when they are challenged, while the latter use succinct cryptographic proof systems to ensure the correctness of the processing of individual or batches of transactions in a short ("succinct") proof. Both archetypes reduce the resource requirements in terms of processing, storage, and bandwidth of full nodes by offloading to another, separate execution layer, while the final execution results are still secured by the Ethereum L1. An example of optimistic rollups is Optimism, which allows the integration of arbitrary Solidity smart contracts. Validity rollups are still mostly in use for relatively simple transactions, such as transfers of ETH and ERC20 tokens. Recently, the first validity rollups were deployed on Ethereum test networks that offer support any Solidity smart contract; these projects are generally termed zk-EVM. Rollups are particularly attractive for mixers, as mixers involve relatively complex cryptographic operations. Mixers have been implemented both on top of optimistic rollups (Tornado Cash fork on Optimism, called Privacy Pools) and validity rollups (e.g., Aztec Connect). Both rollup constructions drastically reduce mixer usage fees. For instance, a deposit transaction that costs around $150 on Tornado Cash costs around $3 when deploying Tornado Cash on Optimism (own measurements, with gas prices from 2023-05-04).

Layer	Dimension	ME	Characteristics				
General features	**Deployment layer**	N	L1	L2 (Optimistic)	L2 (Validity)		
	Governance	N	Single key	Multi-sig	DAO	No on-chain governance	
	Code verification	Y	Audited	Unaudited			
	Contract architecture	Y	Single contract	Multi contract			
Terms of use	**Address registration**	Y	Mandatory	Optional	No registration		
	Mixing usage fee	Y	Fixed	Variable	No fees		
	Usage fee reception	N	(DAO) treasury	Developer(s)	Owner	Liquidity providers	N/A
Mixer functionalities	**Denomination**	Y	Pre-defined	Variable			
	Crypto-asset type	N	ETH	ERC20	ERC721		
	Mixing process	Y	One in, one out	Split and withdraw	Split and use		
	Compliance features	Y	Report generation	None			
Deposit functionalities	**Restitution**	Y	Reversion	Refund	No restitution		
	Deposit censorship	N	Inclusion list	Exclusion list	No censorship		
Withdrawal functionalities	**Execution fee coverage**	N	Relayer	Withdrawer			
	Withdrawal censorship	N	Inclusion list	Exclusion list	No censorship		

Figure 3. Taxonomy of Ethereum-Based Mixers

Governance refers to how a mixer is controlled after deployment. These include control over mechanisms such as contract upgrades, terms of use, and emergency stops, among others. *Single key* refers to singular, centralised control of any type of smart contracts (Zhang et al., 2019) and, thus, also mixers. The key is held by a singular, distinguished entity that has 'owner' privileges. *Multi-sig* is a way to distribute the risks of a singular controlling entity. For instance, a set of keys and a corresponding majority rule can be pre-defined, or a key can be created in a distributed way such that every entity only receives one or multiple sub-keys. A certain threshold of keys is then needed to authorise updates to the smart contract (Komlo & Goldberg, 2021). However, it is hard to determine if these sub-keys were indeed distributed to more than one entity. Another commonly applied governance structure is *decentralised autonomous organisation (DAO)*-based and uses utility tokens that grant voting rights to their holders. Such governance has limitations as the tokens can be distributed unfairly or accumulated by few holders (Barbereau et al., 2023). Because there is only a limited set of rules to be governed in a mixer, there can also be *No on-chain governance framework*.

Code verification considers whether or not the underlying Solidity code-base of a mixer's smart contracts was *Audited* to identify and remove vulnerabilities. The higher the auditor's reputation, the more users may trust a mixer's code base and deployed contract(s). For instance, Cyclon was audited by two auditors (ChainShield & Slowmist, 2023). Some consider audits an essential process to trust in a smart contract's security (Groce et al., 2020). If no such audit takes place, mixers remain *Unaudited*.

Contract architecture refers to the smart contacts underlying the mixer. Mixers can be implemented within a *Single contract* that comprises all of the functionalities. Many others are composed of multiple smart contracts (*Multi contract*). Deploying multiple contracts is mainly motivated by modularity considerations: Each contract can have a single, thus simpler responsibility (such as complex cryptographic operations) and be governed (updated) individually. Some mixers also use individual smart contracts to manage different asset pools (e.g., one for each denomination, see also the **Mixing process**).

Terms of use

Address registration particularly affects some advanced mixers providing direct interfaces to smart contracts associated with decentralised finance (DeFi) protocols. From a technical perspective, creating a new Ethereum account to interact with mixers is not necessary. Hence, many mixers do not provide internal addresses (*No registration*). Some mixers – particularly those that support internal transfers (see also **Mixing process**) – try to popularise their own wallet or reach a better user experience, similar to other DeFi applications that are compatible with wallets like MetaMask. They do so by requiring the registration of a new account that is specifically associated with the mixer (*Mandatory*). The derived account is 'shielded' because it allows users to engage in transfers inside mixers without any linkability between different transactions associated with the same account. In particular, there is no public visibility of the involved account's address. In some cases, registration is supported but not mandatory (*Optional*).

Mixing usage fee corresponds to the transaction commission of a mixer for each deposit of funds. It is a programmable value on top of the transaction fee (which covers the execution of smart contract's functions on the **Deployment layer**). The entity deploying a mixer can define the recipient of the mixing usage fee. Although the recipient could an externally owned account, the mixing fee recipient in all mixers that we investigated and that implement mixing fees is the mixing smart contract itself or another smart contract (see also **Usage fee reception**.. This address makes a profit whenever users mix their crypto assets. The rules to determine the fee are also specified in the mixer smart contract. The fee is usually either flat (*Fixed*) or a small fraction of each deposit transaction (*Variable*), e.g., 0.25 % in RAILGUN. Many mixing contracts feature *No fees*: Owing to the inherent transparency of the blockchain and the publishing of smart contract code to establish trust in mixer usage, any third party can copy the mixer smart contract code, reduce or remove the usage fees, change the recipient, deploy the mixer with these modifications, and overcoming the chicken-and-egg-problem, i.e., trying to attract enough users through the cheaper usage to reach a sufficient level of anonymity guarantees.

Usage fee reception refers to the entity the usage fee is delivered to, e.g., the *DAO treasury* which governs the mixer. It can also be distributed to *Developer(s)* who designed, developed, deployed, and maintain the mixer, to the *Owner* of the mixer smart contract, to *Liquidity providers* that have deposited crypto-assets in

the mixer, or some combination thereof. It would make more sense to reward deposits based on the duration until subsequent withdrawal ("anonymity mining"), but we found no mixer that implemented this approach. *N/A* corresponds to the case where the mixer does not take usage fees.

Mixer functionalities

Denomination refers to which portions of crypto-assets can be deposited and withdrawn by users. Some mixers have a *Pre-defined* set of denominations such that only previously specified denominations of funds can be deposited. For instance, Tornado Cash Nova supports units of 0.1 ETH, 0.3 ETH, 0.5 ETH, and 1 ETH. The fixed denominations are defined at the deployment of the smart contracts. The corresponding deposits and withdrawals can be managed by a smart contract for each denomination or by a single smart contract that comprises all denominations (see **Contract architecture**). For pre-defined denominations, there is a fundamental trade-off between the number of transactions users need to conduct (and pay) to decompose their custom amounts to match the desired amount, and the level of anonymity guarantees: A small variety of denominations can make it inconvenient for users to mix custom amounts; but on the other hand, bundling all deposits into a single pool with one denomination increases the anonymity set. Due to this trade-off, some implementations of mixers are not restricted to certain denominations. Instead, users can deposit and withdraw *Variable* amounts of funds. However, this approach only provides sufficient degrees of anonymisation if crypto-assets can be split inside the pool (see **Mixing process**).

Crypto-asset type refers to the types of assets that can be mixed. The mixers we investigated all target both the native token of the Ethereum blockchain, *ETH*, and the arguably most popular fungible token standard of the Ethereum blockchain, *ERC20*, albeit in different smart contracts. Technically, there is almost no difference between mixing ETH and mixing ERC20 tokens. Implementations compatible with ERC20 involve an additional interaction with the "approve" method of ERC20 smart contracts. As there is no basic depositing method associated with ERC20s, depositors have to authorise that the address of the mixer may spend a certain amount of tokens ("allowance"). Because of the common ERC20 interface, mixers can easily integrate several ERC20 tokens. For instance, Cyclone supports two ERC20 tokens, USDT and TORN. The compatible tokens should be popular enough to have a sufficient number of deposits, otherwise the anonymity set is too small for effective mixing. Because non-fungible tokens (NFTs) are unique (Hartwich et al., 2022), simple mixing is not useful to improve privacy. However, if the mixer supports internal transfers (see **Mixing process**), supporting NFTs is a useful feature. For instance, Nightfall supports the common non-fungible token standard *ERC721*.

Mixing process relates to the operations supported by the mixer. The archetypal one is denoted *One in, one out* because tokens (typically with a fixed denomination) are deposited, pooled, and withdrawn atomically. Other mixers allow for more advanced actions with the deposited funds. For instance, deposited fungible tokens can be split within the system and then sequentially withdrawn (*Split and withdraw*). Alternatively, the deposited amount can be split (again only for fungible tokens) within the system and used for shielded internal transfers and interaction with, for example, DeFi components (*Split and use*). In this case, ownership relations can be changed without the crypto-assets leaving the mixer, such that the mixer provides similar features as a private cryptocurrency like Zcash (for fungible tokens).

Compliance features refer to mixers that enable users to disclose the links between their deposits and withdrawals verifiably. Most commonly, this is achieved through *Report generation*. While their transaction graphs remain obfuscated for observers on the public blockchain, users can provide detailed information to third parties such as auditors and compliance officers. Doing so in confidential communications permits the verification of the provenance of their funds, even though they were sent through a mixer (Nadler & Schär, 2023). While any ZKP-based mixer could offer this functionality (Garman et al., 2017), not all of them provide this functionality to users (*None*).

Deposit functionalities

Restitution applies to mixers with a fixed denomination. When the amount to be deposited is not equal to that denomination, the mixer smart contract either triggers a *Reversion* (the deposit transaction is not

executed and the user wasted transaction fees) or a *Refund* (the difference between the denomination and the deposit is sent back to the depositor). There is *No restitution* in mixers with a variable denomination.

Deposit censorship refers to functionalities allowing the mixer smart contract to incorporate certain aspects of regulatory compliance. One way to do so is to specify an *Inclusion list* or *Exclusion list* that respectively allows or blocks the interaction of specific depositor addresses with a mixer (Burleson et al., 2022). For instance, using an Oracle, mixers can implement exclusion lists to comply with sanction lists, such as the OFAC's list of sanctioned Ethereum addresses. However, sanctions list updates, e.g., after a major attack on a DeFi project, typically happen too slowly to prevent corresponding deposits to the mixer (Burleson et al., 2022). Inclusion lists, on the other hand, can be useful to restrict access to the mixer to addresses that went through a know-your-customer (KYC) process at certain centralised exchanges. There can also be *No censorship*.

Withdrawal functionalities

Execution fee coverage refers to the mechanism by which transaction fees for the smart contract execution layer are paid upon withdrawals. To withdraw deposited funds, users employ cryptographic keys corresponding to the destination address to sign the transaction for invoking the `withdraw` function of the mixer. To make the mixing useful for anonymisation (and as opposed to the case with deposits), the user will generate a new address with no initial funds. The withdrawal transaction requires a fee and potentially a tip to the block producer to incentivise the inclusion of the transaction in a block. Yet, sending funds to the new address from a user's previous address would spoil the obfuscation of the transaction graph. *Relayers* provide a solution to this issue. Users create a transaction that (1) selects a relayer to cover the transaction fee for the `withdraw` function and to withdraw the funds from the mixer, and that forwards the withdrawn amount less the gas and relayer usage fees to the destination address as specified by the user. As such, the user does not need to trust the relayer in terms of confidentiality or crypto-asset custody. Although funding transactions from user-controlled addresses reduces the degree of anonymisation provided by the mixer, many mixers also support fee payment directly by the *Withdrawer*.

Withdrawal censorship refers to the restriction of access to withdrawals. It has the same purpose as for deposit censorship, and *Inclusion lists* play a similar role (e.g., only withdrawals to KYC-ed addresses are allowed). We did not find exclusion lists for withdrawal addresses – which is plausible because they are typically generated only directly before withdrawal. Yet, *Exclusion lists* can be very useful for compliance if they target the depositor address (in *one-in, one-out* mixers). As mixers only provide a high degree of anonymisation when funds are deposited long enough, illicit actors that deposit their funds (e.g., after exploiting a vulnerability in a DeFi protocol) need to be worried that their depositing addresses will be added to the sanction lists that the mixer refers to, which means that they would not be able to access their funds anymore (Burleson et al., 2022). This places a strong incentive for illicit actors not to use such a mixer and, in turn, makes the mixer more attractive for licit users with legitimate privacy needs. As effective mixing relies on the withdrawal transaction not explicitly referring to the corresponding depositing address, the proof of inclusion or exclusion in corresponding lists must be provided by the withdrawer in the form of a ZKP. There can also be *No censorship*.

Evaluation of the Taxonomy

To demonstrate the usefulness and completeness of our taxonomy, we evaluated it following the guidelines by Kundisch et al. (2021). We employed three distinct methods. First, once we observed theoretical saturation in the interviews (after the fourth interview) during our taxonomy-building process, we started to ask the interviewees for their opinions on the usefulness of the taxonomy. The first three interviewees uniformly described the taxonomy as both "functional" for practitioners and "complete". The fourth interviewee stated that there are "no logical flaws", and the fifth and seventh interviewees described the taxonomy as "thorough". Second, we evaluated the taxonomy by applying it to two distinct mixers not used in the taxonomy design phase, RAILGUN and zkBob. Our taxonomy could capture all relevant characteristics of these two mixers. During the evaluation phase, we made a negative observation related to the characterisation of mixers' user experience. Beyond address registration, although user experience aspects may considerably

Layer	Dimension	ME	Characteristics				
General features	Deployment layer	N	L1		L2 (Optimistic)		L2 (Validity)
	Governance	N	Single key	Multi-sig	DAO		No on-chain governance
	Code verification	Y	Audited		Unaudited		
	Contract architecture	Y	Single contract		Multi contract		
Terms of use	Address registration	Y	Mandatory		Optional		No registration
	Mixing usage fee	Y	Fixed		Variable		No fees
	Usage fee reception	N	(DAO) treasury	Developer(s)	Owner	Liquidity providers	N/A
Mixer functionalities	Denomination	Y	Pre-defined		Variable		
	Crypto-asset type	N	ETH		ERC20		ERC721
	Mixing process	Y	One in, one out		Split and withdraw		Split and use
	Compliance features	Y	Report generation		None		
Deposit functionalities	Restitution	Y	Reversion		Refund		No restitution
	Deposit censorship	N	Inclusion list		Exclusion list		No censorship
Withdrawal functionalities	Execution fee coverage	N	Relayer		Withdrawer		
	Withdrawal censorship	N	Inclusion list		Exclusion list		No censorship

Figure 4. Taxonomy Applied to RAILGUN

influence the popularity of this mixer, our taxonomy does not account for pronounced differences in user interfaces. Yet, it is difficult to derive discrete and objective characteristics for such a dimension without a large-scale user study – one that is beyond the scope of our research and may also be difficult to realise owing to the legal issues surrounding mixer usage. Another negative observation was related to the composability feature of RAILGUN. During the application of the taxonomy to RAILGUN (see Figure 4), we noted that it enables direct interactions with other smart contracts associated to, for instance, decentralised exchanges, trading and yield farming, prompting users to keep their assets in the mixer for extended periods. Our taxonomy does not include dimensions that can reflect mixers' capabilities of directly interacting with other smart contracts. However, such features do not describe the intrinsic functionality of mixers. Like user experience aspects, we consider it beyond the scope of our taxonomy. Third, we sent the final taxonomy to all of our interviewees via email, asking for additional written feedback on the finished taxonomy. The first three interviewees reaffirmed its value in written form, describing it as both "useful" and "complete". The fifth and seventh interviewees considered the taxonomy "relevant". Interviewee 6, in consideration of the above-mentioned events related to Tornado Cash, additionally described it as "timely". This evaluation suggests our taxonomy's fitness to cover the key characteristics of Ethereum-based mixers.

Discussion

Our interviews revealed that besides the discrete mixer characteristics we identified when designing the taxonomy, there is a need to consider anonymity guarantees as a pivotal aspect in users' choice of mixers. Because anonymity is multi-faceted and represents a continuum in a highly context- (user base), time- (depositing duration), and user- (expertise of use) specific environment, it is difficult to cover it in a "concise" (Nickerson et al., 2013) way and meet the subjective ending conditions of taxonomy building. To account for this, we discuss this aspect in the following. as well as a critical factor for users' choices (Kruisbergen et al., 2019). Anonymity is a central consideration in both the legitimate (privacy needs) and the illegitimate (e.g., money laundering) use of mixers. Here, anonymity represents an abstraction that considers the anonymity set, i.e., the number of depositors (or depositing transaction) a withdrawal could be associated with, as well as potential scenarios that either improve or degrade the feasibility of 'tracing' funds. Tracing refers to the probabilistic linking of the depositor address and the withdrawal address (or multiple withdrawal addresses for split-and-use systems). The anonymity set of a mixer has a profound impact on the traceability of individual addresses (Wang et al., 2022). For mixers with a fixed denomination, anonymity primarily depends on the number of other users who deposited to the mixer without withdrawing their funds. It is further impacted by the delay between deposit and withdrawal. For mixers without a fixed denomination, the extent of anonymity guarantees depends on the extent to which users split funds on withdrawal and whether it

is impossible to restore the deposited amount by selective addition of the withdrawn amounts. Users with strong privacy needs, such as crime offenders, will arguably favour those mixers with a consistent activity of depositors and a large number of depositors who have not conducted a withdrawal, as this directly impacts the degree of anonymity obtained through mixing. They will also prefer mixers that involve relayers to withdraw mixed assets on a new wallet address with no ETH.

Contributions to theory. Our taxonomy provides an initial but comprehensive classification of non-custodial, non-interactive, Ethereum-based mixers. As such, our research contributes to theory and has implications for prac-tice. It further ex-post demonstrates that the properties of being non-custodial and non-interactive make the design space of Ethereum-based mixers much richer than that of Bitcoin-based tumblers. The design choices we identified (e.g., deployment on L1 or L2 relaying, fee mechanisms, and the implementation of block-lists for depositing and withdrawal) have a profound impact on the functionality offered. Some of these choices may contribute to the extent of their use in criminal activities. As such, our taxonomy stands aside other classifications on crypto-assets and the blockchain ecosystem published in the IS discipline (e.g. Hartwich et al., 2022; Ziegler & Welpe, 2022). These taxonomies, including our own, contribute an understanding of 'fuzzy' and dynamic digital phenomena that – given their greater adoption and decentralised nature – one may characterise as *pervasive* in the sense that mixers erect "novel social formations and behaviours not seen before" (Grover & Lyytinen, 2022, p. 4). Inherently, the value of descriptive classifications lies in their capacity to *explain* (Gregor, 2006). Thus, our contribution to theory is primarily explanatory. Our taxonomy helps researchers by elucidating the task environment or, design choices, of Ethereum-based mixers. It provides an understanding of choices certain user groups, such as crime offenders, can make in the digital age (Kruisbergen et al., 2019). Hence, it represents a contribution to IS research in digital forensics (Nance et al., 2009) and IT artefacts with an inherent dark side (D'Arcy et al., 2014; Tarafdar et al., 2013).

Implications for practice. Together with the Basel Institute on Governance, Europol (2022) advocates for a greater "understanding of crypto-assets and services" as "vital to tackle organised crime and money laundering". Similarly, Interpol (2020) advocates for additional "research to determine the proportions in which criminals prefer [...] mixers over privacy coins". Our work contributes to those ends as it provides an understanding of the most prominent tools used in the money laundering process: mixers. Indeed, such classifications represent valuable analytical instruments for practitioners (Sarre et al., 2018). Consequently, our taxonomy has practical implications for law enforcement and investigations. First, it allows to better understand and identify the various types of Ethereum-based mixers used – both in connection to criminals who launder illicit funds and individuals with legitimate goals – and their implemented as well as conceivable functionalities. Second, by categorising mixers based on their features and characteristics, law enforcement agencies can develop targeted strategies for identifying and tracking illicit transactions. These assist in developing digital investigation instruments and techniques for identifying common patterns or signatures associated with specific types of mixers, which could help them trace funds back to their source or develop an understanding of motives of use (Fröwis et al., 2020). Third, our taxonomy can inform the development of regulations and policies to control mixer usage. By understanding the different ways in which mixers are used, regulators can design more effective measures to prevent and deter money laundering and other criminal activities in the crypto-asset space (see also Barbereau & Bodó, 2023). Lastly, consider that many projects in Web3 aim to protect sensitive information (Lacity et al., 2023), with a significant portion of them using public blockchains, their native crypto-asset, and smart contracts as foundation. Therefore, gaining insights into how certain design choices enable improvements in the protection of sensitive information while also addressing compliance requirements for the case of native cryptocurrency assets could prove highly valuable (Barbereau et al., 2022; Sedlmeir et al., 2022).

Conclusion

Following the taxonomy development steps of Kundisch et al. (2021) and Nickerson et al. (2013), we developed a taxonomy of Ethereum-based, non-custodial, non-interactive crypto-asset mixers. Mixers represent a study object suitable for analysis in research on digital forensics and the dark side of IS (D'Arcy et al., 2014; Tarafdar et al., 2013). We considered both qualitative and quantitative data for the taxonomy development and evaluation over four cycles. As such, the taxonomy we present in this paper is based on (1) a

review of existing peer-reviewed research on mixers, (2) project documentation for four mixers in the form of whitepapers and blog posts, (3) an analysis of the corresponding smart contracts, and (4) expert interviews. We evaluate our taxonomy by applying it to two cases, RAILGUN and zkBob. Our taxonomy provides a comprehensive overview of the novel and previously unstructured research field of Ethereum-based mixers.

Limitations. The limitations of our taxonomy are three-fold. First, we considered a sample of four Ethereum-based mixers to develop our taxonomy and a sample of two for evaluation. These samples offered us rich insights and facilitated the ex-post evaluation of characteristics, dimensions, and layers. Nevertheless, future research could use a larger sample to further evaluate, extend, and potentially modify our taxonomy. Second, while we conducted interviews with experts from different fields to iteratively revise and evaluate our taxonomy, all of them were researchers. Future works could include experts from practice, such as law enforcement agents or software engineers. Third, we are limited by the validity of data used in the initial two (C-2-E) iterations, namely by the reliability of mixers' project documentation. This is a common limitation in taxonomy building (Nickerson et al., 2013). We addressed this issue by critically reflecting on the role and capabilities of the involved technical design choices (e.g., ZKPs) and adding analyses of smart contracts in our data collection.

Future research directions. Considering these limitations, in line with recommendations by Kundisch et al. (2021) and recent related taxonomies (e.g., Hartwich et al., 2022), our taxonomy is intentionally extendable. Taxonomies are by design intended to be encompassing though at the same time allowing for a degree of expansion "when new objects appear" (Nickerson et al., 2013). Ethereum-based mixers are a young and evolving field in research and of relevance to practice. Thus, our taxonomy is designed to flexibly adapt to new developments on all three levels of our taxonomy – layer, dimensions, and characteristics (Nickerson et al., 2013). The close alignment of our taxonomy with the lifecycles of smart contract development and mixing transactions may further increase its flexibility to represent technological advances. Future research could focus on the evaluation and modification of our taxonomy as well as the development of ideal types (i.e., archetypes) (Bailey, 1994) by analysing mixers on smart contract blockchains beyond Ethereum as well as dedicated privacy cryptocurrencies such as Zcash. We also believe that linking research on mixers with research on privacy-oriented and compliant payment systems (such as central bank digital currencies) may be a fruitful endeavour. Research in this area has particularly worked on harmonising privacy requirements with regulatory compliance through cryptographic designs that ensure accountability (e.g., Garman et al., 2017) or that enforce certain transaction limits (Groß et al., 2021). Based on such an extended data set, a joint taxonomy for mixers and privacy pools as well as legal studies on the compatibility of these solutions with KYC and AML requirements could be conducted to continue the corresponding research stream (Barbereau & Bodó, 2023; Barbereau et al., 2022; Trozze et al., 2023).

Despite the limitations discussed above, our analysis and results contribute to both theory and practice. We contribute to the IS discipline's understanding of mixers as pervasive digital phenomena by providing insights into why users with legitimate privacy needs and malicious actors alike may use Ethereum-based mixers and which functionalities they may experience. We provided avenues for future research that may enable a better understanding of users interacting with mixers as well as fill knowledge gaps regarding the different ways to integrate functionalities required for compliance. Our detailed description of the taxonomy's layers and dimensions as well as the corresponding characteristics can support law enforcement in developing digital investigation instruments and targeted strategies to combat illicit finance or proactively provide mixers with compliance features as a legitimate alternative.

Acknowledgements

This research was funded by the Luxembourg National Research Fund (FNR) and PayPal PEARL (grant reference 13342933) as well as by the FNR in the FiReSPArX (grant reference 14783405) and PABLO (grant reference 16326754) projects.

References

Amiram, D., Jørgensen, B. N., and Rabetti, D. (2022). "Coins for bombs: The predictive ability of on-chain transfers for terrorist attacks," *Journal of Accounting Research* (60:2), pp. 427–466. https://doi.org/10.1111/1475-679X.12430.

Andrews, J. (2020). *Aztec: zkRollup Layer 2 + Privacy*.

Bailey, K. D. (1994). *Typologies and taxonomies: An introduction to classification techniques,* SAGE.

Barbereau, T. and Bodó, B. (2023). "Beyond financial regulation of crypto-asset wallet software: In search of secondary liability," *Computer Security & Law Review* (49). https://doi.org/10.1016/j.clsr.2023.105829.

Barbereau, T., Sedlmeir, J., Smethurst, R., Fridgen, G., and Rieger, A. (2022). "Tokenization and regulatory compliance for art and collectibles markets," in *Blockchains and the Token Economy,* pp. 213–236. https://doi.org/10.1007/978-3-030-95108-5_8.

Barbereau, T., Smethurst, R., Papageorgiou, O., Sedlmeir, J., and Fridgen, G. (2023). "Decentralised finance's timocratic governance: The distribution and exercise of tokenised voting rights," *Technology in Society* (73). https://doi.org/10.1016/j.techsoc.2023.102251.

Béres, F., Seres, I. A., Benczúr, A. A., and Quintyne-Collins, M. (2021). "Blockchain is watching you: Profiling and de-anonymizing Ethereum users," in *IEEE International Conference on Decentralized Applications and Infrastructures,* pp. 69–78. https://doi.org/10.1109/DAPPS52256.2021.00013.

Biryukov, A., Feher, D., and Vitto, G. (2019). "Privacy aspects and subliminal channels in zcash," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security,* pp. 1813–1830. https://doi.org/10.1145/3319535.3345663.

Biryukov, A., Khovratovich, D., and Pustogarov, I. (2014). "Deanonymisation of clients in Bitcoin P2P network," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security,* pp. 15–29. https://doi.org/10.1145/2660267.2660379.

Biryukov, A. and Tikhomirov, S. (2019). "Transaction clustering using network traffic analysis for Bitcoin and derived blockchains," in *IEEE Conference on Computer Communications Workshops,* pp. 204–209. https://doi.org/10.1109/INFCOMW.2019.8845213.

Burleson, J., Korver, M., and Boneh, D. (2022). *Privacy-protecting regulatory solutions using zero-knowledge proofs*. a16z crypto.

Buterin, V. (2021). *An incomplete guide to rollups*.

Califf, C. B., Sarker, S., and Sarker, S. (2020). "The bright and dark sides of technostress: A mixed-methods study involving healthcare IT," *MIS Quarterly* (44:2), pp. 809–856. https://doi.org/10.25300/MISQ/2020/14818.

Campbell-Verduyn, M. (2018). "Bitcoin, crypto-coins, and global anti-money laundering governance," *Crime, Law, and Social Change* (69:2), pp. 283–305. https://doi.org/10.1007/s10611-017-9756-5.

Chainalysis (2023). *The 2023 crypto crime report*.

ChainShield and Slowmist (2023). *Audit report by ChainShield and Slowmist*. Cyclone.xyz.

Chan, T. K., Cheung, C. M., and Wong, R. Y. (2019). "Cyberbullying on social networking sites: The crime opportunity and affordance perspectives," *Journal of Management Information Systems* (36:2), pp. 574–609. https://doi.org/10.1080/07421222.2019.1599500.

Cirkovic, M., Cachin, C., and Le, D. V. (2022). "Cryptographic primitives for on-chain tumbler designs,"

Cyclone Community (2023). *Cyclone Protocol: Development*.

D'Arcy, J., Gupta, A., Tarafdar, M., and Turel, O. (2014). "Reflecting on the "dark side" of information technology use," *Communications of the Association for Information Systems* (35:1), pp. 109–118. https://doi.org/10.17705/1CAIS.03505.

Dhillon, G. (2016). "Money laundering and technology enabled crime: A cultural analysis," in *Proceedings of the 22nd Americas Conference on Information Systems,* AIS.

Europol (2022). *Seizing the opportunity: 5 recommendations for crypto assets-related crime and money laundering*.

Feng, Q., He, D., Zeadally, S., Khan, M. K., and Kumar, N. (2019). "A survey on privacy protection in blockchain system," *Journal of Network and Computer Applications* (126), pp. 45–58. https://doi.org/10.1016/j.jnca.2018.10.020.

FIOD (2022). "Arrest of suspected developer of Tornado Cash," *Nieuws*.

Fröwis, M., Gottschalk, T., Haslhofer, B., Rückert, C., and Pesch, P. (2020). "Safeguarding the evidential value of forensic cryptocurrency investigations," *Forensic Science International: Digital Investigation* (33). https://doi.org/10.1016/j.fsidi.2019.200902.

Garman, C., Green, M., and Miers, I. (2017). "Accountable privacy for decentralized anonymous payments," in *Financial Cryptography and Data Security: 20th International Conference,* Springer, pp. 81–98. https://doi.org/10.1007/978-3-662-54970-4_5.

Ghesmati, S., Fdhila, W., and Weippl, E. (2022). "SoK: How private is Bitcoin? Classification and evaluation of Bitcoin privacy techniques," in *Proceedings of the 17th International Conference on Availability, Reliability and Security,* ACM. https://doi.org/10.1145/3538969.3538971.

Goldreich, O. and Oren, Y. (1994). "Definitions and properties of zero-knowledge proof systems," *Journal of Cryptology* (7:1). https://doi.org/10.1007/BF00195207.

Gregor, S. (2006). "The nature of theory in information systems," *MIS Quarterly*, pp. 611–642. https://doi.org/10.2307/25148742.

Groce, A., Feist, J., Grieco, G., and Colburn, M. (2020). "What are the actual flaws in important smart contracts (and how can we find them)?," in *Financial Cryptography and Data Security: 24th International Conference,* Springer, pp. 634–653. https://doi.org/10.1007/978-3-030-51280-4_34.

Groß, J., Sedlmeir, J., Babel, M., Bechtel, A., and Schellinger, B. (2021). *Designing a central bank digital currency with support for cash-like privacy*. https://doi.org/10.2139/ssrn.3891121.

Grover, V. and Lyytinen, K. (2022). "The pursuit of innovative theory in the digital age," *Journal of Information Technology*, pp. 45–59. https://doi.org/10.1177/02683962221077112.

Hartwich, E., Ollig, P., Fridgen, G., and Rieger, A. (2022). "Probably something: A multi-layer taxonomy of non-fungible tokens," *Internet Research*. https://doi.org/10.1108/INTR-08-2022-0666.

Interpol (2020). "Combatting cyber-enabled financial crimes in the era of virtual asset and darknet service providers," *Global Complex for Innovation*.

Komlo, C. and Goldberg, I. (2021). "FROST: Flexible round-optimized Schnorr threshold signatures," in *Selected Areas in Cryptography,* pp. 34–65. https://doi.org/10.1007/978-3-030-81652-0_2.

Kruisbergen, E. W., Leukfeldt, E. R., Kleemans, E. R., and Roks, R. A. (2019). "Money talks money laundering choices of organized crime offenders in a digital age," *Journal of Crime and Justice* (42:5), pp. 569–581. https://doi.org/10.1080/0735648X.2019.1692420.

Kundisch, D., Muntermann, J., Oberländer, A. M., Rau, D., Röglinger, M., Schoormann, T., and Szopinski, D. (2021). "An update for taxonomy designers: Methodological guidance from information systems research," *Business & Information Systems Engineering* (64), pp. 421–439. https://doi.org/10.1007/s12599-021-00723-x.

Lacity, M., Carmel, E., Young, A. G., and Roth, T. (2023). "The quiet corner of Web3 that means business," *MIT Sloan Management Review* (64:3).

Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G. M., and Savage, S. (2013). "A fistful of Bitcoins: Characterizing payments among men with no names," in *Proceedings of the Internet Measurement Conference,* ACM, pp. 127–140. https://doi.org/10.1145/2504730.2504747.

Mikalef, P., Conboy, K., Lundström, J. E., and Popovič, A. (2022). "Thinking responsibly about responsible AI and 'the dark side'of AI," *European Journal of Information Systems* (31:3), pp. 257–268. https://doi.org/10.1080/0960085X.2022.2026621.

Möser, M., Böhme, R., and Breuker, D. (2013). "An inquiry into money laundering tools in the Bitcoin ecosystem," in *APWG eCrime Researchers Summit,* IEEE. https://doi.org/10.1109/eCRS.2013.6805780.

Myers, M. D. and Newman, M. (2007). "The qualitative interview in IS research: Examining the craft," *Information and Organization* (17:1), pp. 2–26. https://doi.org/10.1016/j.infoandorg.2006.11.001.

Nadler, M. and Schär, F. (2023). "Tornado Cash and blockchain privacy: A primer for economists and policymakers," *Federal Reserve Bank of St. Louis Review* (105:2), pp. 122–136. https://doi.org/10.20955/r.105.122-36.

Nance, K., Hay, B., and Bishop, M. (2009). "Digital forensics: Defining a research agenda," in *Proceedings of the 42nd Hawaii International Conference on System Sciences,* ScholarSpace. https://doi.org/10.1109/HICSS.2009.160.

Nickerson, R. C., Varshney, U., and Muntermann, J. (2013). "A method for taxonomy development and its application in information systems," *European Journal of Information Systems* (22:3), pp. 336–359. https://doi.org/10.1057/ejis.2012.26.

Pakki, J., Shoshitaishvili, Y., Wang, R., Bao, T., and Doupé, A. (2021). "Everything you ever wanted to know about Bitcoin mixers (but were afraid to ask)," in *Financial Cryptography and Data Security: 25th International Conference,* Springer, pp. 117–146. https://doi.org/10.1007/978-3-662-64322-8_6.

Pocher, N., Zichichi, M., Merizzi, F., Shafiq, M. Z., and Ferretti, S. (2023). "Detecting anomalous cryptocurrency transactions: An AML/CFT application of machine learning-based forensics," *Electronic Markets* (33:1). https://doi.org/10.1007/s12525-023-00654-3.

Rathore, M. M., Chaurasia, S., and Shukla, D. (2022). "Mixers detection in Bitcoin network: A step towards detecting money laundering in crypto-currencies," in *IEEE International Conference on Big Data,* pp. 5775–5782. https://doi.org/10.1109/BigData55660.2022.10020982.

Ruffing, T., Moreno-Sanchez, P., and Kate, A. (2014). "Coinshuffle: Practical decentralized coin mixing for Bitcoin," in *Proceedings of the 19th European Symposium on Research in Computer Security,* Springer, pp. 345–364. https://doi.org/10.1007/978-3-319-11212-1_20.

Sánchez-Gómez, N., Torres-Valderrama, J., García-García, J. A., Gutiérrez, J. J., and Escalona, M. (2020). "Model-based software design and testing in blockchain smart contracts: A systematic literature review," *IEEE Access* (8), pp. 164556–164569. https://doi.org/10.1109/ACCESS.2020.3021502.

Sarre, R., Lau, L. Y.-C., and Chang, L. Y. (2018). "Responding to cybercrime: Current trends," *Police Practice and Research* (19:6), pp. 515–518. https://doi.org/10.1080/15614263.2018.1507888.

Schultze, U. and Avital, M. (2011). "Designing interviews to generate rich data for information systems research," *Information and Organization* (21:1). https://doi.org/10.1016/j.infoandorg.2010.11.001.

Sedlmeir, J., Lautenschlager, J., Fridgen, G., and Urbach, N. (2022). "The transparency challenge of blockchain in organizations," *Electronic Markets* (32), pp. 1779–1794. https://doi.org/10.1007/s12525-022-00536-0.

See, K. (2023). "The Satoshi laundromat: A review on the money laundering open door of Bitcoin mixers," *Journal of Financial Crime*. https://doi.org/10.1108/JFC-11-2022-0269.

Shen, M. (2022). "Crypto mixer Tornado Cash says sanctions can't apply to smart contracts," *Bloomberg*.

Tarafdar, M., Gupta, A., and Turel, O. (2013). "The dark side of information technology use," *Information Systems Journal* (23:3), pp. 269–275. https://doi.org/10.1111/isj.12015.

Tornado Cash Team (2021). *Tornado Cash introduces arbitrary amounts & shielded transfers*.

Trozze, A., Davies, T., and Kleinberg, B. (2023). "Of degens and defrauders: Using open-source investigative tools to investigate decentralized finance frauds and money laundering," *Forensic Science International: Digital Investigation* (46). https://doi.org/10.1016/j.fsidi.2023.301575.

TyphoonCash (2021). *Introducing Typhoon Cash – A new protocol for yield-capable private transactions*.

U.S. Department of the Treasury (2022). *U.S. Treasury sanctions notorious virtual currency mixer Tornado Cash*.

U.S. Department of the Treasury (2023). *Treasury designates Roman Semenov, co-founder of sanctioned virtual currency mixer Tornado Cash*.

van Wegberg, R., Oerlemans, J.-J., and van Deventer, O. (2018). "Bitcoin money laundering: mixed results? An explorative study on money laundering of cybercrime proceeds using Bitcoin," *Journal of Financial Crime* (45:2), pp. 419–435. https://doi.org/10.1108/JFC-11-2016-0067.

Wang, Z., Chaliasos, S., Qin, K., Zhou, L., Gao, L., Berrang, P., Livshits, B., and Gervais, A. (2022). "On how zero-knowledge proof blockchain mixers improve, and worsen user privacy," in *Proceedings of the ACM Web Conference,* pp. 2022–2032. https://doi.org/10.1145/3543507.3583217.

Zhang, R., Xue, R., and Liu, L. (2019). "Security and privacy on blockchain," *ACM Computing Surveys* (52:3). https://doi.org/10.1145/3316481.

Zheng, P., Zheng, Z., Wu, J., and Dai, H.-N. (2021). "On-chain and off-chain blockchain data collection," in. *Blockchain Intelligence: Methods, Applications and Challenges,* pp. 15–39. https://doi.org/10.1007/978-981-16-0127-9_2.

Ziegler, C. and Welpe, I. M. (2022). "A taxonomy of decentralized autonomous organizations," in *Proceedings of the 43rd International Conference on Information Systems,* AIS.