Linking NFT Transaction Events to Identify Privacy Risks

Dorottya Zelenyanszki[1(✉)], Zhé Hóu[1(✉)], Kamanashis Biswas[2(✉)], and Vallipuram Muthukkumarasamy[1(✉)]

[1] Griffith University, Brisbane, Australia
{dorottya.zelenyanszki,z.hou,v.muthu}@griffithuni.edu.au
[2] Australian Catholic University, Brisbane, Australia
kamanashis.biswas@acu.edu.au

Abstract. Non-fungible tokens (NFTs) are unique tokens with various domains, e.g. real estate, metaverse, gaming and public auctions. However, when minted on public blockchains, the underlying blockchain transaction data can be publicly accessible. This instigated transaction data analysis for various purposes, including cryptocurrency price prediction and NFT market analysis. The public data may be considered privacy-sensitive which sets a barrier to the wider adoption of NFTs. In this work, we present that the analysis of the transaction events can describe activities in NFT applications by establishing connections between transactions and thereby, it can identify information that may be privacy-sensitive. This can be useful in developing suitable privacy-enhancing methods for NFTs. We collected transaction data from a blockchain-based game called Planet IX that was built on the Polygon blockchain and used graph visualisation to provide examples for constructed connections.

Keywords: blockchain · non-fungible tokens · privacy · data analysis

1 Introduction

Public blockchains provide a decentralised and secure environment to build decentralised applications (dApps). Those applications often involve NFTs, which introduce multiple use cases for these tokens, such as metaverse objects, game items, art and tickets [10]. As NFTs can represent any unique item, the number of application areas may potentially grow. Public blockchains are accessible by anyone, and transaction data has been presented to be utilised to analyse blockchain activities [11]. Transactions include multiple pieces of information that can be considered privacy sensitive, such as wallet addresses and transacted values, and it can also be linked to other transactions and from that, even more connecting data can be extracted. This can lead to a number of potential privacy issues such as deanonymization, transaction fingerprinting or transaction pattern exposure [3].

Several technologies such as zero-knowledge proofs (ZKPs) [5–9] and differential privacy [8] have been applied to enhance the privacy of blockchain-based

N. Dong et al. (Eds.): SDLT 2023, CCIS 1975, pp. 82–97, 2024.
https://doi.org/10.1007/978-981-97-0006-6_6

applications. The identification of the privacy-critical NFT-related information can enable the development of improved, more application-specific privacy-enhancing techniques because a more detailed picture of the application's privacy situation is provided.

Previous studies also conducted research on blockchain transaction analysis, including data related to NFTs. However, these mainly focus on network evolution analysis [13], anomaly/vulnerability detection [4], cryptocurrency price prediction [6] or NFT market analysis [2]. However, there is a potential for dApps that involve social interactions, such as multiple players in games or interactions between avatars in the virtual worlds of metaverses. NFTs are highly suitable for these types of applications as they can represent the users but also the digital objects they are interacting with.

In the submitted transactions of the dApps, multiple events are also emitted. These events also include information that potentially can be sensitive or can be utilised to link multiple events or transactions, and by that, additional information can be revealed, which then can lead to the construction of behavioural patterns of the dApp's user base. Their analysis, therefore, can lay down a foundation for identifying the privacy-sensitive NFT-related data which then can prompt the introduction of updated privacy-preserving methods. Although, our focus is the NFTs, the analysis of the events is general and not restricted to our NFT scope.

In order to conduct the analysis, we collected transactions from a blockchain-based game that includes NFTs called Planet IX[1] which runs on the Polygon PoS[2], which is an Ethereum scalability solution. We extracted the basic transaction information and the event logs, and we categorised both the event types and properties. We also used graph visualisation to show how events can connect to other events in different transactions. Finally, we also discussed how the combination of these can be utilised for the detection of privacy-critical information.

The rest of the paper is structured as follows: Sect. 2 introduces related works to this research. Following that, in Sect. 3 we give a high-level overview of the concept that describes how we analyse blockchain transaction logs. In Sect. 4 we explain how the data collection has been conducted. In Sect. 5 we discuss in detail how we analysed the data, present some visualisation results and describe what the results can be used in regards to NFT privacy. Finally, in Sect. 6 we conclude the paper and mention our planned future steps to enhance this research.

2 Related Works

This section presents the related research works. It provides examples of both blockchain-based analysis and previously used privacy-enhancing techniques and also describes where this research offers an enhancement in this area.

Zhao et al. [13] described the Ethereum blockchain as an ecosystem that consists of users and contracts that cohabit with the blockchain fabric. It is not like

[1] https://planetix.com/.
[2] https://polygon.technology/.

an online social network or a financial network; it is more like the Internet where users and programs interact with each other based on predefined rules. They aimed to study Ethereum by examining all interactions (user-to-user, user-to-contract, contract-to-user, contract-to-contract) in order to explore the evolution of the network, its properties and communities. To achieve this, they constructed four temporal networks from Ethereum and they applied global network properties to detect changes and anomalies. They also leveraged machine learning models to make predictions regarding the continuation of the determined communities. They presented that these techniques can be applied in areas such as blockchain intelligence and blockchain-based social networks.

Hu et al. [4] stated that classification could help identify smart contract vulnerabilities because contracts have different behavioural characteristics and application use cases, which show a variance in their detection. The classification can potentially also rely on the deployer of a contract because it can reveal the true purpose of the contract and it can also consider the identified design issues because they can potentially consume a large amount of gas. For this purpose, they manually analysed 10,000 smart contracts. They identified 4 behaviour patterns, and 14 basic features and also designed a data-slicing approach to minimalise the negative effect of insufficient datasets, which enabled them to present the effectiveness of the approach in an LSTM network.

Casale-Brunet et al. [2] mentioned that there can be a parallel drawn between NFT transaction graphs and graphs that are used to describe social media interactions. The latter has been previously used to determine user preferences, and they stated that there is a possibility that related algorithms can be leveraged to identify trusted/influential wallets and analyse market evolution. To explore this area, a systematic analysis has been conducted on the evolution of the NFT communities based on their interaction graphs and related properties. This analysis presented results in identifying so-called super nodes, which are wallets that coexist in multiple NFT collections and that have been presented to be influential on the market.

Wan et al. [9] mentioned that smart contracts take off-chain data as input through interactions. They also added that it is highly important to provide data authenticity and privacy protection for the off-chain data. Their research on existing works presented that they only offer a solution for either of those; therefore, to provide an enhancement on this, they designed an extended zero-knowledge succinct non-interactive argument of knowledge (zk-SNARK) called zk-DASNARK that handles the data authentication by also leveraging digital signatures. Their model, a zero-knowledge authenticated data feed system (zk-AuthFeed) utilises zk-DASNARK to provide data authentication and privacy protection for dApps.

Huang et al. [5] focused on data availability in decentralised storage such as blockchain. They proposed a data integrity checking protocol. This protocol uses efficient verifiable delay functions (EVDF), Fiat-Shamir ZKPs, Merkle trees and smart contracts to offer this functionality. This way, they presented a protocol

that does not leak important information, ensures fairness among participants and provides public verification.

These works showed that blockchain-related data analysis may include research on the evolution of communities, but it does not cover in-depth analysis of the user activities that are occurring in the dApps. We state that the events emitted in blockchain transactions can describe user activities within the applications, including giving a picture of the situation regarding the privacy-critical information that is shared. This can be facilitated by introducing similar upgraded privacy-preserving techniques to [5–9] that are based on specific privacy-sensitive information identified by the transaction events analysis. It can also lay down a foundation for user behaviour analysis by leveraging techniques used for similar purposes in social networks. Similar to how Adali et al. [1] constructed new behavioural features to understand user behaviour on Twitter or how Yang et al. [12] utilised factor graph model to make predictions in regards to retweeting behaviour.

Table 1. Basic collected features

Name	Description
blockHash	Unique block identifier
blockNumber	Number of the block the transaction occurred in
transactionHash	Unique transaction identifier
timestamp	Date and time
from	Sender address
to	Receiver address
value	Value of the transaction
isError	Whether any error occurred during execution (boolean)
txreceipt status	Transaction execution status message
contractAddress	Smart contract address
methodId	Transaction method identifier
input	Transaction input data

3 Overview of the Proposed Concept

This section presents the proposed concept for identifying privacy-critical information. It is divided into three phases: data collection, visualisation through graphs and analysis, and the concept can be seen in Fig. 1. In the following, we describe each step and also refer to the section where they are described in detail.

3.1 Data Collection Phase

At first, transaction data has to be extracted from the application's underlying blockchain through publicly available APIs. The dApps have one or multiple

smart contracts that handle their activities. Therefore, for sufficient data collection, it is advised to extract data from multiple contracts. Need to highlight that it is also important to collect data that belongs to the same timeframe as this data going to be leveraged to describe user activities over time later. From that transaction data, basic features (detailed in Table 1) that are already used in blockchain analysis (e.g. wallet addresses) are extracted, and the transaction's event logs are also decoded. This data is then converted into a CSV format which is suitable for establishing the graph visualisation later. The data collection phase is described in detail in Sect. 4.2.

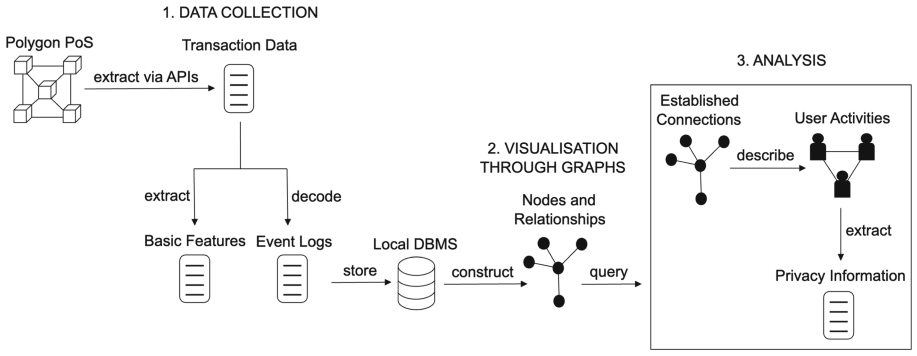

Fig. 1. Overview of the proposed method.

3.2 Visualisation Through Graphs Phase

From the converted CSV files, multiple graph nodes and relationships are constructed as described in Sect. 5.1. These are then stored in a local database. Using the nodes and relationships the collected data can be queried to establish connections between distinct blockchain data that would not be connected by default. We provided examples for these in Sect. 5.2. These examples show how a single event property can link multiple transactions including ones that happened at different time points as they belong to differing blocks. If an event property type and value pair has the ability to establish connections to a high number of events that belong to other transactions, then we can consider it privacy-critical. Identifying these critical pairs is the first step towards using this concept to describe the privacy of NFTs.

3.3 Analysis Phase

These connections can be leveraged to describe user activities within the application over time which can eventually also reveal the privacy information that is leaked through events and has to be protected through specifically designed novel privacy-enhancing techniques as the analysis of these connections gives a

picture of the application from a privacy point of view. We also plan to use these connections to construct general privacy patterns so we can leverage established techniques from social network analysis to add a more general classification of privacy-sensitive NFT-related data in dApps. These patterns can describe certain NFT activities or attacks.

4 Data Collection and Processing

In order to present the significance of the concept proposed in the previous section, a small data collection and analysis have been conducted. In this section, we describe the tools we utilised for this purpose and also present how the data collection was performed.

Table 2. Event types

Type	Description	Example
log	Log events of application activities	LogFeeTransfer
mint	Mint events	PIXMinted
NFT	Certain NFT-related events	NFTPlaced
token	Events that cover tokens	TokenClaimed
transfer	Transfer events	Transfer
staking	Staking related events	TokenStaked
other	Every other type of events e.g. application-specific events	Combined

Table 3. Event property type groups

Group	Description	Example
app	App-related properties	operator, approved
log	Properties of log events	input1, output1
user	Properties with address value	account, user
id	Id properties	pixId
token	Token-related properties	tokenId, tokenAddress
value	Value properties	amount
staking	Staking properties	stakeable
other	Other properties e.g. application-specific events	location (x, y)

4.1 Experimental Setup

We conducted the experiment on macOS 13.2.1 and used Python 3.9.6 to run scripts to extract blockchain transaction data and convert it into a suitable CSV

format. For analysis and visualisation, the Neo4j Desktop 1.5.8[3] was utilised where we established a local DBMS 1.5.8 to store the data that can be used for this. To perform queries, we also added the extended APOC library[4].

4.2 Data Collection

Transaction data has been extracted from three smart contract addresses of Planet IX. The first one is the contract for the PIX NFT[5] which is a hexagon-shaped virtual copy of a part of the planet Earth that users can own in this game. The second contract is for Mission Control[6] where the users can stake their NFT assets such as PIX NFTs. The third and final contract is for Gravity Grade[7], which is an in-game corporation through which the users can obtain fresh PIXs [7]. We collected 1000 transactions from each address between the block numbers 43845596 and 46574022, thereby enabling us to see user activities within the same time period. In the future, more transactions can be extracted from these addresses, and more addresses from the game can be involved to present a picture of user activities on a larger time period and on a wider scale of the application.

To obtain transactions and their events, we used PolygonScan API[8] and the Polygon PoS API on Alchemy[9]. For each transaction, some basic features have been extracted, and the event logs have also been decoded. The basic features can be seen in Table 1. Since every included smart contract address uses the EIP-1967 Transparent Proxy pattern[10], we had to obtain the implementation address first in order to get the correct contract ABI, which is required to decode the event logs. Both the basic features and the decoded logs for each transaction were placed into a JSON file. We also extracted each address and implementation address and put it into a separate JSON file so we could use it later for filtering purposes.

After this step, we went through the transactions again and checked whether they had any decoded event logs associated with them. If yes, we categorised each event into one of the event types presented in Table 2, and we also grouped all the event properties into groups presented in Table 3. The types and the groups were assigned based on the event and property names. This is the reason for having both NFT and token event types; as for the latter, it is not certain that it is NFT-related because the game includes other types of tokens as well. Although it can be assumed to be the same NFT token. We plan to use these types and groups for the analysis of other applications as well; therefore, eventually, they are going to be generalised. We also assigned unique IDs for both the events and

[3] https://neo4j.com/.
[4] https://neo4j.com/labs/apoc/5/.
[5] https://polygonscan.com/address/0xb2435253c71fca27be41206eb2793e44e1df6b6d.
[6] https://polygonscan.com/address/0x24e541a5c32830a4e8b89846fd4bf86e294dd3cb.
[7] https://polygonscan.com/address/0x3376c61c450359d402f07909bda979a4c0e6c32f.
[8] https://docs.polygonscan.com/.
[9] https://docs.alchemy.com/reference/polygon-api-quickstart.
[10] https://eips.ethereum.org/EIPS/eip-1967.

their event property type and value pairs so they can be easily queried later. We also collected every unique event property type and value pair separately as they are often repeated in subsequent events. We extracted three separate CSV files from this process: one for the events, one for all the event property types and value pairs and one for every unique event property type and value pairs.

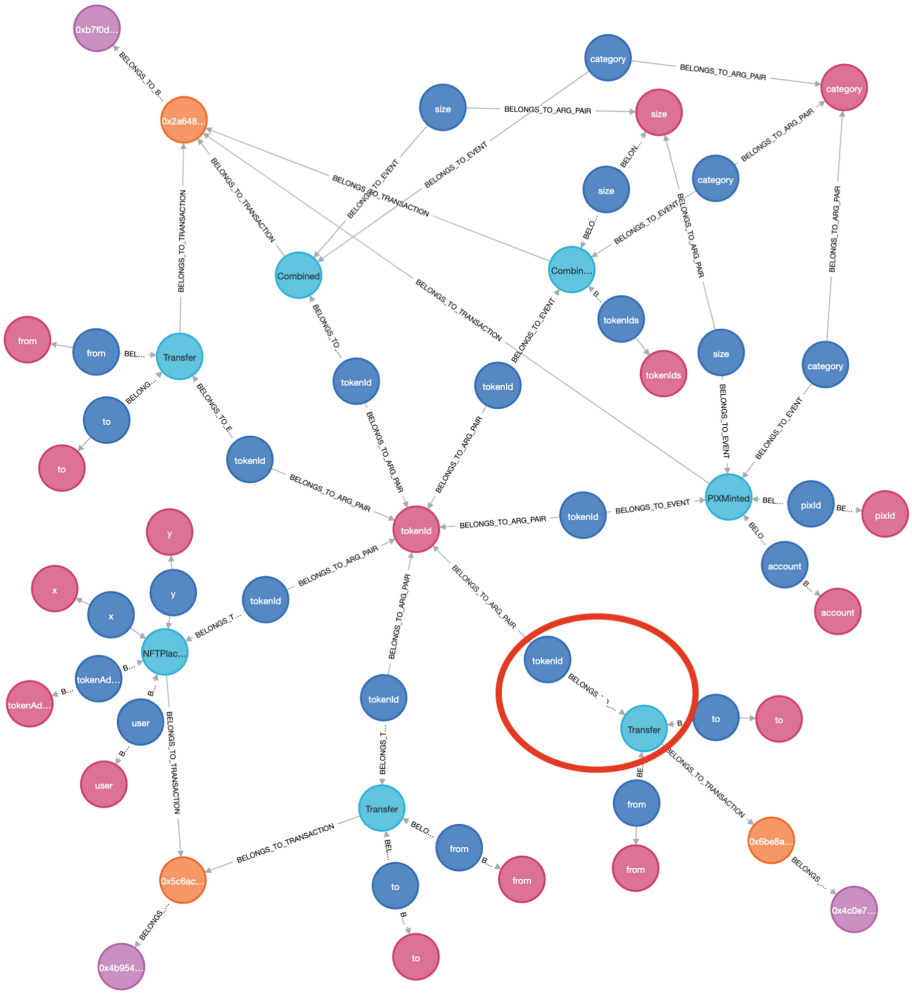

Fig. 2. Linking transactions based on one event which is highlighted with the red circle (Color figure online)

5 Analysis and Discussion

The collected, categorised data in CSV format enabled us to visualise the user activities in a graph format. This not only allowed us to present them in an easy-to-understand form but also presented how to connect multiple pieces of information initiated from only one event. In this section, we describe how we established the DBMS in Neo4j and present the results in the discussion section.

Table 4. Number of connecting events per event property type and value pair group

Group	# connecting events
user	3012534122
other	371951778
id	58806468
token	17231980
value	1106028
staking	774080

Table 5. Number of connecting events per event type

Type	# connecting events
transfer	3011980562
mint	224203980
other	216506850
nft+token	9713064
staking	0

5.1 Neo4j Setup

In order to visualise the data from the data collection, we introduced 6 types of nodes and 4 types of relationships based on the 4 types of CSV files for each contract address. Note that events with a log type and property type and value pairs with an app or log group have been discarded during the import to the DBMS. The log events and properties have been excluded because they do not offer new information. They usually log event data that has been previously emitted through other events such as when an approval event is emitted after a transfer event. The app properties have been neglected because our focus is on user activities that can lead to identifying privacy-sensitive information that relates to the NFTs. We describe them as follows:

Nodes:

1. Block(blockHash, blockNumber, timeStamp): This refers to the blocks in the blockchain that can be identified by the unique block hash or by the blockNumber. They include multiple transactions, and they also determine the date and time for all those transactions by the timestamp. Blocks are represented by purple colour.
2. Transaction(transactionHash, blockHash, fromAddr, toAddr, methodId, value): This refers to the transactions in the blockchain identified by their transaction hashes. It also has block hashes as properties to identify which block the transactions belong to. The other properties are a subset of the basic

features from Table 1. Transactions can emit events as well. Transactions are represented by orange colour.
3. Event(uuid, event, type, transactionHash): This node describes the events that are emitted through transactions that are determined by their assigned unique IDs. It also includes properties for the name of the event and its categorised type. The transaction the event belongs to can be extracted by the included transaction hash. Events are represented by light-blue colour.
4. Arg(uuid, argType, argValue, group, eventUuid): This node determines all the event property type and value pairs. They are all identified by a preassigned unique ID and also include a property for the group they belong to. The event in which they have been emitted is determined by the eventUuid property. Property pairs are represented by dark-blue colour.
5. ArgPair(argType, argValue, group): Some event property types and value pairs are repeating across multiple events from differing transactions and blocks. This node refers to every unique pair. Unique property pairs are represented by red colour.
6. Contract(address): This node describes every contract and implementation address that has been deducted from decoding the events. These nodes are used for filtering. Contracts are represented by green colour, although, they are never part of the resulting graph of the query.

Relationships:

1. BELONGS_TO_BLOCK: This relationship returns with transaction and block pairs where the transaction has been submitted to the blockchain within that particular block.
2. BELONGS_TO_TRANSACTION: This relationship returns with event and transaction pairs where the event has been emitted through that particular transaction.
3. BELONGS_TO_EVENT: This relationship returns with the event property type and value pair and event pairs where the event property belongs to that particular event.
4. BELONGS_TO_ARG_PAIR: This presents how event property type and value pairs can repeat throughout multiple events. The relationship shows that every pair belongs to a unique pair.

5.2 Visualising Through Graph Format

The established nodes and relationships can be utilised to query the collected data. Note that the results of the queries have been limited in order to present graphs that have a number of nodes and edges that make the resulting graphs still visually pleasing but also show how can we link NFT-related information.

Case Study: Location Data Leakage. In Fig. 2 we present an example of how multiple events, transactions and blocks can be connected even through

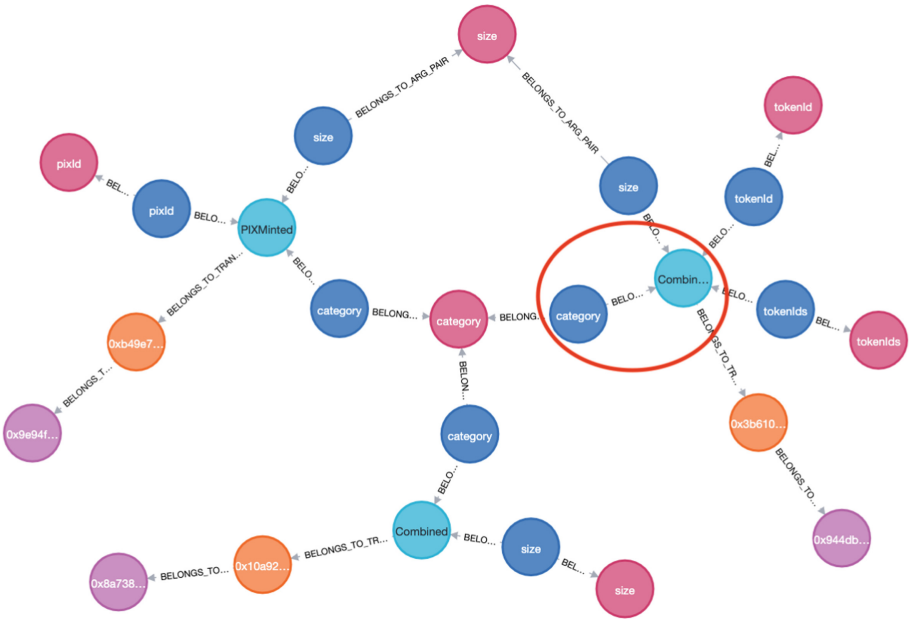

Fig. 3. Linking transactions through an event property pair that belongs to the other group (Color figure online)

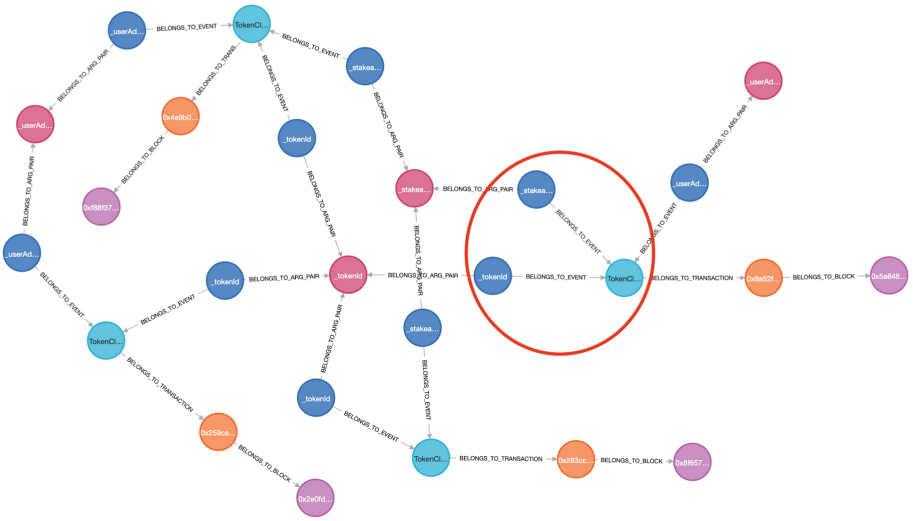

Fig. 4. Linking transactions based on an event that belongs to the token type (Color figure online)

one event. We take a simple Transfer event and its tokenId property (highlighted with the red coloured circle). By using the BELONGS_TO_ARG_PAIR relation, we identify the unique property pair (displayed at the centre of the figure) that connects the first event to other events. In this case, this pair is the previously mentioned tokenId event property and its value. By leveraging the other relations, we also present that with this query we cover three points in time because the connecting transactions belong to three different blocks. This suggests that through the analysis of the event logs, we can present what sort of activities the event property pairs are involved in over time. This can be utilised for various types of use cases including providing NFT life-cycle information. We also displayed every event's other properties and which unique pairs they belong to. This can reveal additional information about the highlighted event property pair. For example, the connecting NFTPlaced event has the x and y properties, which gives location information for that particular NFT in the game at that particular point in time. This NFT is identified by the token address included in the NFTPlaced event and by the tokenId through which the two events are connected. Information revealed in this way can be potentially privacy-sensitive. For example, in a metaverse setting a piece of similar location information can reveal where the participant's NFT avatars are located within the virtual world, which can be highly useful for malicious actors who try to introduce behavioural patterns of victim users so they can commit user-specific malevolent actions. The query to construct this graph is the following:

```
MATCH (c:Contract)
MATCH r1=(a1:Arg)-[:BELONGS_TO_EVENT]->(e1:Event), r2=(
    a1)-[:BELONGS_TO_ARG_PAIR]->(p:ArgPair), r3=(e1)-[:
    BELONGS_TO_TRANSACTION]->(t1:Transaction), r4=(t1)
    -[:BELONGS_TO_BLOCK]->(b1:Block), r5=(a2:Arg)-[:
    BELONGS_TO_EVENT]->(e1), r6=(a2:Arg)-[:
    BELONGS_TO_ARG_PAIR]->(), r7=(a3:Arg)-[:
    BELONGS_TO_ARG_PAIR]->(p), r8=(a3)-[:
    BELONGS_TO_EVENT]->(e2:Event), r9=(a4:Arg)-[:
    BELONGS_TO_EVENT]->(e2), r10=(a4)-[:
    BELONGS_TO_ARG_PAIR]-(), r11=(e2)-[:
    BELONGS_TO_TRANSACTION]->(t2:Transaction), r12=(t2)
    -[:BELONGS_TO_BLOCK]->(b2:Block) WHERE NOT a1 = a2
    AND NOT a3 = a4 AND NOT e1 = e2 AND NOT toLower(p.
    argValue) = toLower(c.address)
RETURN r1,r2,r3,r4,r5,r6,r7,r8,r9,r10,r11,r12
LIMIT 400
```

Filtering Using Types and Groups. We can also use the event types and event property pair groups to provide similar graphs to show the linking of information. For example, in Fig. 3 we see how we can connect events through an event that has an event property pair that is part of the other group which

probably means that it is an application-specific property. In that figure, we highlighted the initiating CombinedWithBurned event and the connecting category property pair with a red coloured circle. In Fig. 4 we take an event that specifically belongs to the token type. This results in a graph where multiple TokenClaimed events connect through multiple property pairs. Within the red circle, we can see how events either connect through the tokenId or through the stakeable property pair from the initiating event's point of view. But two of them also connect via another separate userAddress property pair as well. These types of queries can be leveraged to describe the privacy influence level of certain events and property groups which can eventually help us filter out certain events that have negligible importance in describing user activities. The queries for these graphs are identical to the previously listed query, but they have an additional condition for either the type or the group. For example, the query for Fig. 3 is as follows:

```
MATCH (c:Contract)
MATCH r1=(a1:Arg)-[:BELONGS_TO_EVENT]->(e1:Event), r2=(a1)-[:
    BELONGS_TO_ARG_PAIR]->(p:ArgPair), r3=(e1)-[:
    BELONGS_TO_TRANSACTION]->(t1:Transaction), r4=(t1)-[:
    BELONGS_TO_BLOCK]->(b1:Block), r5=(a2:Arg)-[:BELONGS_TO_EVENT
    ]->(e1), r6=(a2:Arg)-[:BELONGS_TO_ARG_PAIR]->(), r7=(a3:Arg)-[:
    BELONGS_TO_ARG_PAIR]->(p), r8=(a3)-[:BELONGS_TO_EVENT]->(e2:
    Event), r9=(a4:Arg)-[:BELONGS_TO_EVENT]->(e2), r10=(a4)-[:
    BELONGS_TO_ARG_PAIR]-(), r11=(e2)-[:BELONGS_TO_TRANSACTION]->(t2
    :Transaction), r12=(t2)-[:BELONGS_TO_BLOCK]->(b2:Block) WHERE
    NOT a1 = a2 AND NOT a3 = a4 AND NOT e1 = e2 AND NOT toLower(p.
    argValue) = toLower(c.address) AND p.group = "other"
RETURN r1,r2,r3,r4,r5,r6,r7,r8,r9,r10,r11,r12
LIMIT 80
```

5.3 Discussion

In order to understand what NFT-related information is privacy-sensitive, at first, we have to determine what type of information has the ability to establish multiple connections that can describe user activities and eventually user behaviour which can then enable us to leverage techniques that are used in social networks for user analysis.

The event types and event property pair groups can be leveraged for this purpose. Through a simple query, we can check which type and group has the highest number of connecting events. Note that at this stage we take the NFT and token events together as there is a need for additional data in order to determine how to differentiate them completely. Table 4 and 5 present that in general events are connected through a pair that belongs to either the user or other group and the connecting events are usually transfer, mint or other events. As the staking group provided the least connecting pairs and there were no staking events that made a connection, we can assume that staking does not have a significant influence on determining user activities which means that it can be potentially filtered out.

Although this presents the user group and the transfer events as major connectors, when we look into the event properties of the connecting events, we can see that the user, other and token groups are all able to make connections to event properties from five different groups. The exact numbers can be seen in Table 6. We assume that the underwhelming influence of the user group comes from the fact that 6981 events out of 8518 total events are transfer events that usually involve wallet address properties that belong to the user group. This proves that events can be connected through multiple types of properties and via these connecting event property types and value pairs we can associate information with information that automatically would not be assumed. For example, it is clear that an NFT has a tokenId and an owner just by using the underlying smart contract; however, application-specific information such as its category, ID or location may not be part of its metadata (which can be extracted by the contract) or transactions events submitted by its contract. Therefore, an event property that has high connection ability can be declared as information that has a high influence on privacy as it will establish connections to a high number of differing event properties which enhances the richness of the described user activities which then enables us to eventually reveal an increased number of privacy-critical NFT-related information. In order to lay down the foundation for user activity analysis, more application data has to be extracted from the blockchain as we need a greater variety of events so we can establish event property groups and event types that are more specific.

Table 6. Number of connecting events categorised by the group of their connecting property pair

Group	# connecting events	
user	user	1505161855
	token	1505532427
	id	609708
	value	249852
	other	980280
other	user	41746605
	token	165294690
	id	41045067
	staking	54096
	other	123811320
token	user	1111067
	token	5436129
	id	4368
	staking	53256
	other	10627160
id	user	14801472
	token	14641704
	value	79884
	other	29283408
value	user	737352
	id	368676
staking	token	387040
	other	387040

6 Conclusion

The lack of privacy of the NFTs is one of the major obstacles to applying them on a wider application scale. The information revealed through transactions (e.g. addresses, value) can be considered privacy-sensitive; therefore, it needs some type of privacy-enhancing technique to be utilised to protect it. We also argue that the event logs emitted in the transaction can cause further privacy leakage and it can be also utilised to link multiple transactions. In this research, we

presented how transactions from a chosen application can be extracted and then how their events can be decoded and utilised to establish connections between information pieces that by definition may not be connected. We argue that by establishing these connections, user activities of dApps can be extracted which is the first step to identifying the privacy-critical NFT-related information in dApps which can eventually lead to the establishment of user behavioural patterns. We also used graph-based visualisation to present examples of the connections. The analysis of blockchain transaction events can also enable the development of novel privacy-enhancing techniques that are based on the identified privacy-critical NFT-related data.

In our future work, we will collect more data from the already mentioned smart contracts and include additional contracts from the application. This will enable us to refine the event types and groups and show a wide variety of user activities in a longer time period. User activities may be analysed to identify the privacy-sensitive information. This may reveal various types of issues and vulnerabilities in regard to the NFTs and their end-users. By leveraging already established social network analysis techniques we also plan to introduce general privacy patterns which may then be utilised to introduce effective methods for privacy protection. More experiments may be performed with a different dApp to validate the results from the first instalment.

References

1. Adali, S., Golbeck, J.: Predicting personality with social behavior. In: 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 302–309 (2012). https://doi.org/10.1109/ASONAM.2012.58
2. Casale-Brunet, S., Ribeca, P., Doyle, P., Mattavelli, M.: Networks of Ethereum non-fungible tokens: a graph-based analysis of the ERC-721 ecosystem. In: 2021 IEEE International Conference on Blockchain (Blockchain), pp. 188–195. IEEE Computer Society, Los Alamitos (2021). https://doi.org/10.1109/Blockchain53845.2021.00033
3. Feng, Q., He, D., Zeadally, S., Khan, M.K., Kumar, N.: A survey on privacy protection in blockchain system. J. Netw. Comput. Appl. **126**, 45–58 (2019). https://doi.org/10.1016/j.jnca.2018.10.020, https://www.sciencedirect.com/science/article/pii/S1084804518303485
4. Hu, T., et al.: Transaction-based classification and detection approach for Ethereum smart contract. Inf. Proc. Manage. **58**(2), 102462 (2021). https://doi.org/10.1016/j.ipm.2020.102462, https://www.sciencedirect.com/science/article/pii/S0306457320309547
5. Huang, Y., Yu, Y., Li, H., Li, Y., Tian, A.: Blockchain-based continuous data integrity checking protocol with zero-knowledge privacy protection. Digit. Commun. Netw. **8**(5), 604–613 (2022). https://doi.org/10.1016/j.dcan.2022.04.017, https://www.sciencedirect.com/science/article/pii/S2352864822000669
6. Ozer, F., Sakar, C.O.: An automated cryptocurrency trading system based on the detection of unusual price movements with a time-series clustering-based approach. Expert Syst. Appl. **200**, 117017 (2022). https://doi.org/10.1016/j.eswa.2022.117017, https://www.sciencedirect.com/science/article/pii/S0957417422004353

7. Software, N.: Whitepaper (2022). https://planetix.gitbook.io/whitepaper/. Accessed 3 Aug 2023
8. Ul Hassan, M., Rehmani, M.H., Chen, J.: Differential privacy in blockchain technology: a futuristic approach. J. Parallel Distrib. Comput. **145**, 50–74 (2020). https://doi.org/10.1016/j.jpdc.2020.06.003, https://www.sciencedirect.com/science/article/pii/S0743731520303105
9. Wan, Z., Zhou, Y., Ren, K.: zk-AuthFeed: protecting data feed to smart contracts with authenticated zero knowledge proof. IEEE Trans. Dependable Secure Comput. **20**(2), 1335–1347 (2023). https://doi.org/10.1109/TDSC.2022.3153084
10. Wang, Q., Li, R., Wang, Q., Chen, S.: Non-fungible token (NFT): overview, evaluation, opportunities and challenges (2021)
11. Wu, J., Liu, J., Zhao, Y., Zheng, Z.: Analysis of cryptocurrency transactions from a network perspective: an overview. J. Netw. Comput. Appl. **190**, 103139 (2021). https://doi.org/10.1016/j.jnca.2021.103139, https://www.sciencedirect.com/science/article/pii/S1084804521001557
12. Yang, Z., et al.: Understanding retweeting behaviors in social networks. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM '10, pp. 1633–1636. Association for Computing Machinery, New York (2010). https://doi.org/10.1145/1871437.1871691
13. Zhao, L., Sen Gupta, S., Khan, A., Luo, R.: Temporal analysis of the entire Ethereum blockchain network. In: Proceedings of the Web Conference 2021, WWW '21, pp. 2258–2269. Association for Computing Machinery, New York (2021). https://doi.org/10.1145/3442381.3449916