# Supporting Criminal Investigations on the Blockchain: A Temporal Logic-based Approach

Marco Blanchini[1,2], Michele Cerreta[1], Davide Di Monda[1,3], Matteo Fabbri[1,4], Mario Raciti[1,5], Hamza Sajjad Ahmad[1] and Gabriele Costa[1]

[1]*IMT School for Advanced Studies, Lucca, Italy*
[2]*University of Siena, Siena, Italy*
[3]*University of Napoli Federico II, Naples, Italy*
[4]*University of Florence, Florence, Italy*
[5]*University of Catania, Catania, Italy*

**Abstract**

Over recent years there has been a growing focus on cyber-enabled crimes within criminal investigations. The rising trend of Ransomware-as-a-Service (RaaS) highlights a pattern where criminal groups explicitly demand cryptocurrency payments from their victims and then distribute these funds to their affiliates. This strategy leverages key-blockchain characteristics such as anonymity, decentralization, transparency, and immutability of transactions to evade regulatory oversight and hinder digital investigation efforts, while simultaneously building trust among affiliates. This paper proposes an approach to query the blockchain to support criminal investigations using temporal logic. It features an oracle that allows investigators to perform detailed queries and analyze specific properties of transactions or addresses within the blockchain. The implementation is achieved via a Python engine, which includes a query language, an interpreter, and a client interface. The practical application and effectiveness of our approach are illustrated through a real-world case study, highlighting its utility.

**Keywords**

Cryptocurrency forensics, Ransomware, Formal language, Cybercrime, Money laundering detection

## 1. Introduction

The era of cryptocurrencies started in 2008 when Satoshi Nakamoto first launched the *Bitcoin*.[1] Since then, the attention toward cryptocurrency and distributed ledger technologies has grown exponentially. One of the main reasons is their decentralized structure which leads to several benefits that include economic efficiency, transparency, anonymity, absence of a central power, and financial inclusion [1]. The flip side is that these features are also extremely appealing for criminal businesses as they facilitate operations like money laundering, tax evasion, illicit payments, regulation avoidance, and various types of frauds [2]. For instance, cryptocurrencies are the default currency for *ransomware attacks*. These phenomena have prompted increased attention from law enforcement agencies and governments. As a consequence, they started

*ITASEC 2024: The Italian Conference on CyberSecurity*
marco.blanchini@imtlucca.it (M. Blanchini); michele.cerreta@imtlucca.it (M. Cerreta); davide.dimonda@imtlucca.it (D. D. Monda); matteo.fabbri@imtlucca.it (M. Fabbri); mario.raciti@imtlucca.it (M. Raciti); hamza.sajjadahmad@imtlucca.it (H. S. Ahmad); gabriele.costa@imtlucca.it (G. Costa)


[1]https://bitcoin.org/en/

international initiatives for enhancing the control and regulations over the blockchain and, thus, for countering the blockchain-related cybercrimes [3].

Furthermore, cryptocurrencies not only pose challenges for law enforcement but also offer unprecedented opportunities for investigative efforts. The decentralized and transparent nature of blockchain technology enables forensic analysts to trace the flow of funds and unravel complex financial networks employed by cybercriminals [4]. Investigators can follow the money trail, identify key nodes, and uncover patterns indicative of criminal activity. For instance, we might consider the case of *Ransomware-as-a-Service* (RaaS) [5]. As of 2023, more than 72% of businesses on a global scale experienced the deleterious effects of ransomware attacks[2]. In this context, analyzing transactional data on the blockchain can reveal insights into the distribution of ransom payments, the modus operandi of ransomware operators, and potential strategies employed to evade detection [2, 6].

However, this requires advanced network analysis techniques. Therefore, it becomes essential to develop novel techniques and tools to analyze underlying properties and identify recurring patterns within the distributed ledger. In this paper, we propose an *approach to query the blockchain in support of criminal investigations leveraging temporal logic.*

### 1.1. Structure Summary

The rest of the paper is organized as follows. In Section 2, we provide the information needed to foster the need for this research effort from a criminological point of view. Section 3 discusses the related works, with a particular focus on the hot topic related to tracking movements of Bitcoins resulting from illicit activities. Section 4 delves into the technical aspects of our approach, elucidating its methodology, implementation, data sources, and analytical capabilities. Section 5 applies the approach to a real-world case study. Finally, Section 6 concludes the paper, drawing the main remarks and future directions.

## 2. Background and Motivation

Criminal investigations in the context of blockchain technology, particularly concerning cryptocurrencies like Bitcoin, pose unique challenges for law enforcement agencies worldwide. Taking as example RaaS, investigations are hindered by the difficulty in identifying perpetrators: indeed, despite the considerable growth of these crimes, the cases in which investigative activities have been successfully concluded are limited.

The main investigative approach to date consists in tracking the path of payments made by victims through cryptocurrencies [7]: identifying the origin of such transactions, however, is often infeasible because of the anonymity granted to their authors by the blockchain. In this regard, Yousaf [7] advances the idea of exploiting the organizational characteristics of RaaS groups to trace their transactions. In fact, surveys and recent criminological studies [8] have shown that these organizations have two types of members: "core" members run the organization, while "affiliate" members support it in criminal activities. The "affiliate" members

[2]Annual share of organizations affected by ransomware attacks worldwide from 2018 to 2023, Statista, 2023, https://www.statista.com/statistics/204457/businesses-ransomware-attack-rate/

of the organization are scattered in different countries and usually do not know each other or the "core" members.

Analyses of the few arrests made by the authorities, such as the captures of Mikhail Vasiliev[3] in 2022 and Ruslan Astamirov in 2023 [4] belonging to the *Lockbit* group, confirm these organizational characteristics. This evidence suggests that RaaS groups probably need to divide ransom payments to "affiliate" members in a systematic manner without making the cryptocurrency route traceable. The very high frequency of criminal activities also underlines that the redistribution may be operated by automated methods on the blockchain. Consequently, we hypothesize that these specific requirements lead RaaS perpetrators to construct cryptocurrency paths that exhibit particular regularities and transactional structures. Based on the argument presented earlier, this paper introduces the following *research question*:

> RQ. *Can we query the blockchain and extract data in an aimed way, using fine-grained and customizable rules?*

Following this research question, our study progresses in the direction of building an approach based on *formal language to detect regularities and potential patterns in the blockchain*, intending to help law enforcement officials trace illicit transactions and identify their authors.

## 3. Related Work

The necessity to identify and track illicit money flows transiting through the blockchain has sparked significant research interest in recent years. With the rise of cryptocurrencies and their decentralized nature, traditional methods of monitoring financial transactions have proven inadequate, necessitating innovative approaches. To achieve this goal, researchers have explored a diverse range of methodologies, leveraging advanced technologies such as Machine Learning (ML) and Deep Learning (DL). These techniques offer promising avenues for detecting and preventing malicious activities within the blockchain ecosystem. In Table 1, we compile and organize the most recent and notable works that present techniques aimed at addressing malicious phenomena in the blockchain.

Notably, all the works in the literature span from 2018 to 2023. An high concentration of paper can be found in 2023. This confirms the recent interest of the research community. Beyond recency, we specifically selected papers that focus on identifying and tracing patterns associated with such activities. This ensures that our study aligns with the current research landscape and addresses its key objectives.

The first column of the table details the *focus* of the papers. Particularly, the most common problem that the state-of-the-art faces is tracking Bitcoin related to *ransomware* attacks [11, 13, 14, 18]. Other works [9, 12, 15] offer a broader scope depending on the type of classes (viz. *specific illegal activities*) contained in the dataset used in the proposed analyses. Additionally,

[3] Man Charged for Participation in LockBit Global Ransomware Campaign, Office of Public Affairs, 2023, https://www.justice.gov/opa/pr/man-charged-participation-lockbit-global-ransomware-campaign

[4] Russian National Arrested and Charged with Conspiring to Commit LockBit Ransomware Attacks Against U.S. and Foreign Businesses, Office of Public Affairs, 2023, https://www.justice.gov/opa/pr/russian-national-arrested-and-charged-conspiring-commit-lockbit-ransomware-attacks-against-us

**Table 1**
Related studies focusing on the detection of money flows in the blockchain related to illegal activities. Papers are arranged chronologically by publication year. The final row provides an overview of the present work. The meaning of acronyms is provided at the bottom of the table.

| Paper | Year | Focus | Methodology | Query |
|---|---|---|---|---|
| Harlev et al. [9] | 2018 | Illicit activities | ML | ✗ |
| Bartoletti et al. [10] | 2018 | Ponzi schemes | ML | ✗ |
| Huang et al. [11] | 2018 | Ransomware | E2E An | ✗ |
| Sun Yin et al. [12] | 2019 | Illicit activities | ML | ✗ |
| Akcora et al. [13] | 2019 | Ransomware | TDA+ML | ✗ |
| Kappos et al. [14] | 2022 | Ransomware | Cl+He | ✗ |
| Nerurkar [15] | 2023 | Illicit activities | DL | ✗ |
| Yu et al. [16] | 2023 | Money laudering | DL | ✗ |
| Pocher et al. [17] | 2023 | Money laudering | DL | ✗ |
| Ampel et al. [18] | 2023 | Ransomware | DL | ✗ |
| *This work* | 2024 | Pattern detection | FL | ✓ |

**Focus**—Main focus of the proposed methodology; **Query**— Whether the methodology allows querying the blockchain in real-time or not; **Acronyms**: Machine Learning (ML), Deep Learning (DL), Formal Language (FL), E2E An (End-to-end Analysis), TDA+ML (Topological Data Analysis with Machine Learning), Cl+He (Clustering and Heuristic methodology).
✓ Present. ✗ Absent.

another topic explored in the literature involves *money laundering* operations in the blockchain, as examined in [16, 17]. Lastly, Bartoletti et al. [10] focus on the detection of *Ponzi scheme*. None of the devised works introduces a method that is general enough to be applicable to various regularities and patterns on the blockchain. Conversely, the focus is limited to a set or to one particular scenario.

In the *methodology* column, we detail the proposed techniques to address the aforementioned issues. Notably, the majority of the works adopt an AI-based approach, and both ML and DL models have been explored. Two works [9, 12] cluster the addresses contained in the considered datasets and employ supervised ML models (e.g., Gradient Boosting, Random Forest, K-Nearest Neighbors) to classify the clusters based on the particular illicit activity that characterizes them. Bartoletti et al. [10] apply data mining techniques to detect Bitcoin addresses related to Ponzi schemes. First, they constructed a dataset by analyzing the blockchain transactions used to perform the scams. Then, they used this dataset to train various ML algorithms, assessing their effectiveness. The authors conclude that the best classifiers can identify most of the Ponzi schemes in the dataset, with a low number of false positives. An ML-based framework has also been proposed in [13] for detecting malicious addresses associated with ransomware groups. In their work, the authors introduce the concept of topological data analysis (TDA) to detect ransomware patterns on the blockchain. The intuition is that it could be possible to extract hidden data patterns with a systematic analysis of data shapes, such as cycles and flares, quantified at various resolution scales.

DL models have also been recently used in different papers [15, 16, 17, 18]. In detail, the

majority of them [15, 17, 16] leverage Graph Neural Networks (GNN) for detecting anomalous activities. Similarly, Ampel et al. [18] design a framework for labeling nodes in ransomware payment networks. Such a framework is based on a semi-supervised GNN jointly used with *GraphSAGE* to handle class imbalance.

Regarding non-AI techniques, Huang et al. [11] discuss the feasibility of tracking ransomware payments over a two-year span, providing a data-driven approach to understanding the operational mechanisms of ransomware campaigns. Kappos et al. [14] deal with the validation and expansion of Bitcoin clusters. The authors present a heuristic for identifying peel chain[5] transactions on the blockchain.

Although AI-based methodologies offer considerable potential, they also face hurdles. The most prominent limitation of ML and DL models is that they often *require retraining* when new criminal activity patterns emerge. This can create significant scalability and maintenance challenges. Furthermore, explaining how AI models (especially DL, given its black-box nature) reach their conclusions remains a major roadblock. This *lack of transparency* is especially critical for forensic activities, where clear and explainable results are required for legal proceedings.

Regarding the last column, it is evident that in the current state-of-the-art, there are no solutions capable of directly *querying* the blockchain to check whether specific properties are met. While various techniques such as ML and DL have shown promise in detecting suspicious activities within the blockchain, they often rely on analyzing transaction patterns, network structures, or other data attributes rather than directly querying the blockchain for specific properties. This limitation underscores the challenges in real-time monitoring and enforcement of compliance measures within decentralized systems.

### 3.1. Positioning

After presenting the state-of-the-art in the previous section, in the last row of Table 1, we summarize the positioning of this work. Several studies [9, 12, 10, 11] rely on ML methods for analysis, whereas others [15, 16, 17, 18] leverage more complex DL techniques. Differently from the reviewed literature, our work focuses on using a *formal language*-based approach for querying the blockchain. The reasons behind our choice to utilize formal language are presented below:

- The ever-increasing use of black-box AI models in cybersecurity applications calls for a focus on explainability and transparency, principles that undergird the requirements of the upcoming *AI Act* of the European Union. The use of AI needs to be directed by a human-in-the-loop approach that integrates human stakeholders in the ML/DL pipeline. Capuano et al. [19] clarify that, in the context of cybersecurity, "explainability can only occur via human-machine interaction": indeed, the system should provide "responses to queries that users are likely to ask", especially for tasks that require the traceability of transactions and perpetrators. Most research on explainable AI for cybersecurity[6] focuses on post-hoc explanations, which do not allow one to understand the rationale

[5]The peel chain method involves the laundering of significant volumes of illicitly acquired cryptocurrency through a series of numerous small transactions.

[6]It is worth noting that explainability modules are not present in the reviewed literature dealing with AI techniques.

of the detection process in real time [19]. Instead, the querying approach we propose is intrinsically *explainable*, as it is not based on black-box AI models.

- Formal language-based querying offers enhanced *adaptability* to changes in behaviors within transactions, ensuring the relevance of our approach in dynamic environments. Specifically, it allows for the identification of any type of pattern expressible through the language.
- Unlike ML/DL techniques, formal language querying typically consumes *fewer computational resources*, including processing power and memory. This efficiency makes our approach more practical and feasible, particularly in environments with limited computational resources or where real-time processing is essential.

After outlining the advantages of our approach, we employ basic temporal logic rules to create a formal language designed for querying the blockchain. This process aims to improve the precision and interpretability of our analysis of blockchain transactions, leading to better identification and understanding of transaction history.

## 4. Approach

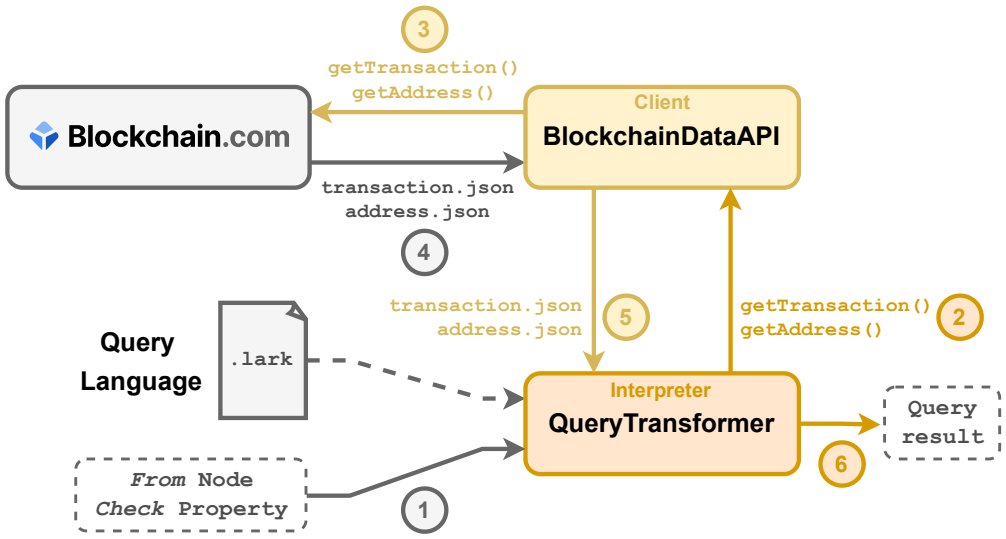


**Figure 1:** Overview of the proposed approach and its main components.

This paper proposes an approach that relies on basic temporal logic rules to build a formal language to query the blockchain. Temporal logic offers a formal framework to express and evaluate complex temporal properties over time, thus enabling the detection of trends, anomalies, and emergent behaviors within blockchain transactions. In fact, in the case of malicious activities on the blockchain, it is necessary to evaluate the characteristics of transactions chains that span over time via ad-hoc temporal operators. The main contribution of our approach consists in an

oracle that allows to perform queries to check properties for a given transaction or address in the blockchain.

The oracle is implemented as a *Python* engine that consists of three main components:

- **Query Language** – It provides the syntax for the expression of the queries.
- **Interpreter** – It parses and interprets the queries and links the query language to the client for the evaluation of the rules.
- **Client** – It retrieves the actual data from the blockchain, leveraging *Blockchain.com*[7].

While the code is available open source in a repository[8] and each component is detailed below, Figure 1 depicts an overview of the structure and interaction within the three components. Furthermore, Figure 1 illustrates the flow of information and the exchange of requests/responses between the different modules when a query is made.

### 4.1. Query Language

Our approach relies on an expressive query language. Thus, the first step for our approach is the definition of the query language through the following grammar.

$$\varphi ::= \varphi_t \mid \varphi_w$$
$$\varphi_t ::= P(a_t) \mid \neg\varphi_t \mid \varphi_t \wedge \varphi_t \mid \varphi_t \vee \varphi_t \mid X_w\varphi_w \mid F_w\varphi_w \mid G_w\varphi_w$$
$$\varphi_w ::= P(a_w) \mid \neg\varphi_w \mid \varphi_w \wedge \varphi_w \mid \varphi_w \vee \varphi_w \mid X_t\varphi_t \mid F_t\varphi_t \mid G_t\varphi_t$$

Briefly, the grammar includes propositional logical operators as well as linear temporal operators [20]. A query is a formula $\varphi$ that can either be a transaction formula $\varphi_t$ or a wallet address (or address in short) formula $\varphi_w$. Both transaction and address formulas share the same structure, i.e., a formula can be an atomic predicate on a transaction/address attribute, in symbols $P(a_t)$ and $P(a_w)$, respectively. For instance, an atomic predicate on a wallet's balance may have the form: balance $>$ 3 BTC. Then, both $\varphi_t$ and $\varphi_w$ can contain negations ($\neg$), conjunctions ($\wedge$), disjunctions ($\vee$), and temporal operators. Temporal formulas require more attention. As a matter of fact, since the blockchain consists of alternating transactions and wallet addresses, temporal operators must also be alternating. Thus, we label traditional temporal operators $X$ (next), $F$ (eventually) end $G$ (globally) to define their scope, i.e., either transactions or addresses. For instance, a transaction formula $\varphi_t$ may include $G_w$, while an address formula $\varphi_w$ may include $X_t$.

**Example 1.** *Consider the formula $\varphi_t = X_w X_t$ sent $<$ 10 BTC. It is satisfied by the addresses, being destinations of an initial transaction (see below), such that their next outgoing transaction is worth less than* 10 *BTC.*

The semantics of a formula must be evaluated against an entry node of the correct type, i.e., a transaction for $\varphi_t$ and an address for $\varphi_w$. Intuitively, the evaluation of a formula $\varphi$ against entry node $s$ corresponds to exploring the blockchain until $\varphi$ holds. The explored fragment of

[7] https://www.blockchain.com/explorer/api
[8] https://github.com/IMTAltiStudiLucca/blockchain-ransomware

the blockchain is then returned. More formally, the semantics of a formula $\varphi$ (under node $s$), in symbols $s \vdash_{t/w} \varphi$, is defined as follows.

$s \vdash_{t/w} P(a)$ iff $P(a)$ holds in $s$
$s \vdash_{t/w} \neg\varphi$ iff $s \nvdash_{t/w} \varphi$
$s \vdash_{t/w} \varphi \wedge \varphi'$ iff $s \vdash_{t/w} \varphi$ and $s \vdash_{t/w} \varphi'$
$s \vdash_{t/w} \varphi \wedge \varphi'$ iff $s \vdash_{t/w} \varphi$ or $s \vdash_{t/w} \varphi'$
$s \vdash_{t/w} X_{t/w}\varphi$ iff $s \to s'$ and $s' \vdash_{w/t} \varphi$
$s \vdash_{t/w} F_{t/w}\varphi$ iff $\exists s'.s \to^* s'$ and $s' \vdash_{w/t} \varphi$
$s \vdash_{t/w} G_{t/w}\varphi$ iff $\forall s'.s \to^* s'$ implies $s' \vdash_{w/t} \varphi$

From the above operators, we can derive the following instances for transactions and addresses: `XTrans`, `FTrans`, `FAddr`, `GTrans`, and `GAddr`. Generally, unbounded queries are not practical, hence we introduce an upper limit $n$ to achieve a reasonable balance. Furthermore, we enrich the set of operators with the round parentheses for the prioritization of multiple properties and with the `MaxAddr` custom operator. In summary, we define the following operators:

- `XTrans`: Specifies that a condition must hold true at the next transaction.
- `FTrans` $n$: Specifies that a condition must eventually become true in a future transaction within $n$ steps.
- `FAddr` $n$: Specifies that a condition must eventually become true in a future address within $n$ steps.
- `GTrans` $n$: Specifies that a condition must hold true for all $n$ future transactions.
- `GAddr` $n$: Specifies that a condition must hold true for all $n$ future addresses.
- `MaxAddr`: Specifies that a condition must hold true for the address having the maximum amount of a given attribute.

The entry point for the formula is given by the "`query`" rule, and the grammar expects queries in the following format: "`From node Check property`", where *node* can be either a transaction and/or a wallet address, and *property* is the condition to be checked. This can be either a transaction property or an address property. A transaction/address *expression* represents basic expressions in the form of algebraic comparisons, like equal, greater than, less than, between atomics attributes (of a transaction/address).

## 4.2. Interpreter

The second component of our approach is the interpreter. It provides a link between the query language and the client. In particular, we defined the query language using *Lark*[9], which is a *Python* library for parsing custom languages. Therefore, the language, as previously defined, is written with the syntax proposed by *Lark*—based on *Extended Backus–Naur form* (EBNF)—in a `.lark` file. Parsing is carried out via the *LALR* (short for "look-ahead, left-to-right") parsing algorithm. The parsed query is then transformed in an *Abstract Syntax Tree* (AST), which is then evaluated through a *Transformer* class. Such class is implemented in our framework with the "`QueryTransformer`" class, which encompasses a set of callbacks that are executed for each

[9] https://github.com/lark-parser/lark

branch of the AST. Callbacks implement the logic needed to put in place the queries (e.g., the actual property checking, the requests to be made to the blockchain interface, etc.).

The interpreter includes sanity checks to ensure that the types of elements passed in a query are consistent with the types that the query language would expect. Furthermore, the interpreter follows a linear structure, i.e., it implements a method corresponding to each non-terminal rule defined in the grammar. For the sake of brevity, we only describe the core of the interpreter here. The core is implemented in the "`_prop_checker`" callback, which acts as the main *query evaluation* method. Briefly, it parses the operators used in a query and evaluates the properties. In the case of temporal operators like `GTrans`, `FTrans` et similia, the "`_prop_checker`" method calls itself recursively to check the property in the subsequent transaction/addresses. These are retrieved by calling the client, as it will be described later. It follows naturally that the "`_prop_checker`" callback handles each representation of transaction/address property in a different yet appropriate way. The method takes multiple arguments as input, depending on the query, and a boolean to differentiate between transactions and addresses. The implementation of the checks for `XTrans`, `GTrans`, `FAddr`, and `MaxAddr`, are still a work in progress, thus the default return value set to "False". For the sake of brevity, the code is here omitted, while it is still available online.

### 4.3. Client

The third component of our approach is the (blockchain) client. It relies on the free available plan from Blockchain.com API service to retrieve the data as JSON outputs directly from the Bitcoin blockchain. The logic is implemented in the "`BlockchainDataAPI`" class, which provides basic methods for the retrieval of block, transaction, and address data. While the code is, again, available online, we focus our attention on the information that is retrievable from the APIs. These APIs offer rich information, enabling the application of the approach across various use cases and allowing for the verification of various transaction and address properties. For instance, it is possible to retrieve information such as wallet balance, number of inputs and outputs, lock time, fees, etc., which can be useful for investigative purposes.

## 5. Demo

In this section, we apply our approach to a real-world case study, that is, the *Colonial Pipeline* attack[10]. First, we give a little context to the case study, especially from a criminalistics perspective. Then, we formulate a query related to the pipeline case to confirm the operation and effectiveness of our approach in support of criminal investigations.

On the 7th of May 2021, Colonial Pipeline, a vital supplier of energy to the southeastern United States, experienced a crippling ransomware attack. The assault, orchestrated by the Russia-based cybercriminal group *DarkSide*, compelled Colonial to halt its operations temporarily. In a swift response to the crisis, Colonial paid a ransom of 75 Bitcoin—equivalent to approximately $4.4 million—to the attackers. Despite this payment, the pipeline remained offline for six days, exacerbating concerns over fuel shortages in affected regions. Panic buying ensued as news of the

[10] https://en.wikipedia.org/wiki/Colonial_Pipeline_ransomware_attack

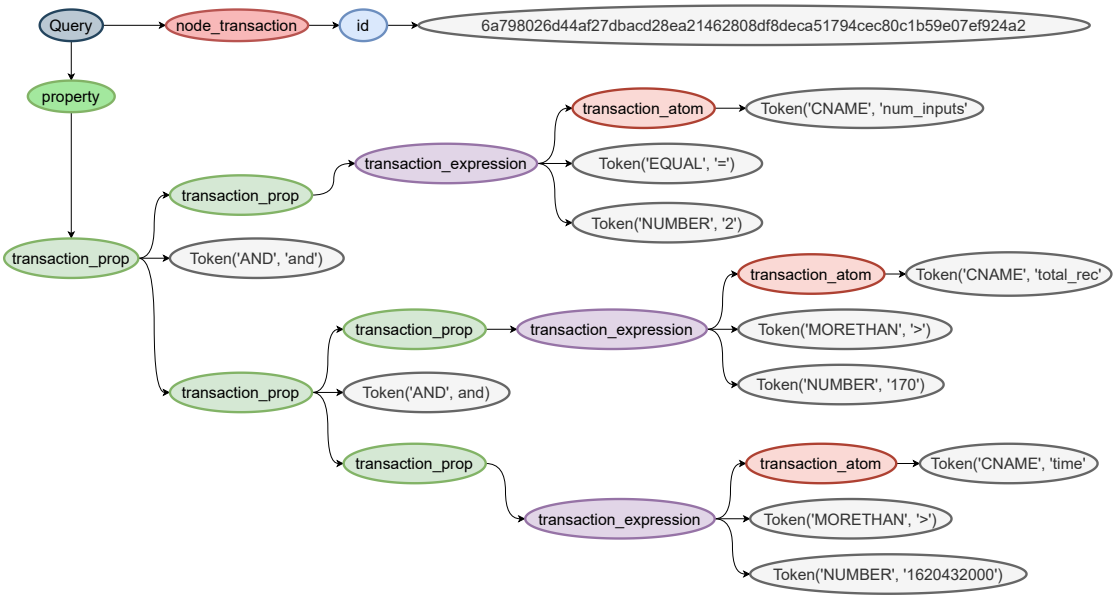


**Figure 2:** AST for the Colonial Pipeline query.

disruption spread, compounding the challenges faced by consumers and authorities alike. After the investigations into the pipeline case, the techniques of money laundering and cryptocurrency redistribution by ransomware organizations have significantly evolved. In particular, the *Conti* group, starting from 2019, began employing more intricate and effective blockchain money laundering techniques, which have also been adopted by other major ransomware groups [21]. Furthermore, these sophisticated money laundering techniques often do not require high technical expertise but can be carried out using tools readily available on the internet and used with relative ease. For such reasons, combating this criminal phenomenon requires improving both quantitative and qualitative analysis techniques of transactions on the blockchain [22].

Below, we set a simple query that analyses the first transaction related to the aforementioned case study. Given relevant information, such as the date of the attack and the ransom demanded, from the pipeline case, it is possible to formulate the following query.

```
From Transaction 6a798026d44af27dbacd28ea21462808df8deca51794cec80c1b59e07ef924a2 Check
Transaction.num_inputs = 2 and Transaction.total_rec > 170 and Transaction.time > 1620432000
```

This query allows us to verify our investigation hypotheses. These include the number of inputs of the transaction under examination (viz. `Transaction.num_inputs`), which must be equal to 2 according to the hypotheses, the total Bitcoin amount requested (viz. `Transaction.total_rec`), which is greater than 170 BTC, and the date of the transaction following the ransomware attack (viz. `Transaction.time`)—this is known to be after the 8th of May 2021 0 AM. The execution of the query yields the AST depicted in Figure 2. As a result, the query is evaluated with a return value “True”, thus confirming that the transaction under investigation matches the hypothesized patterns.

## 6. Conclusions

The development of tools to analyze transactions on the blockchain is one of the current priorities to counter cybercrime. Nowadays, the ability to launder money on the blockchain is one of the greatest strengths of cybercriminals: this advantage allows them to act with impunity worldwide. In fact, tracing the origin of illicit money is one of the pillars of investigative activity and crime fighting. However, this investigative approach is currently infeasible for law enforcement agencies, and the accuracy of the analyses by cybercrime scholars is also limited. The principal step in this direction is to develop the ability to filter illicit financial transactions by recognizing their characteristics among all the transactions in the blockchain.

This paper addressed the challenge of formalizing the process of querying the blockchain. It answered the research question by advancing a querying approach that represents the foundation for developing tools that can have a twofold benefit in countering cybercrime: (*i*) on the one hand, they could allow law enforcement officials to identify illicit transactions on the blockchain with a certain probability; (*ii*) on the other hand, they could improve scholars' and analysts' ability to understand money laundering techniques and systems on the blockchain through more reliable data on the phenomenon. Furthermore, the impact of the proposed approach was demonstrated by applying it to a real-world case study with promising results.

Arguably, the *Python* engine still needs to be refined and completed, in particular for the implementation of the temporal operators, yet it has already proved to work for quite complex queries. Ultimately, our future work also looks at the development of a novel framework that can be leveraged for several scopes, such as tracing ransomware payments conducted through cryptocurrency transactions.

### Acknowledgments

This work was partially supported by project SERICS (PE00000014) under the NRRP MUR program funded by the EU – NGEU.

## References

[1] C. Patsakis, E. Politou, E. Alepis, J. Hernandez-Castro, Cashing out crypto: state of practice in ransom payments, International Journal of Information Security (2023) 1–14.

[2] M. D. He, M. K. F. Habermeier, M. R. B. Leckow, M. V. Haksar, M. Y. Almeida, M. M. Kashima, M. N. Kyriakos-Saad, M. H. Oura, T. S. Sedik, N. Stetsenko, et al., Virtual currencies and beyond: initial considerations, International Monetary Fund, 2016.

[3] J. Sakellariadis, Behind the Rise of Ransomware, Technical Report, Atlantic Council, 2022. URL: http://www.jstor.org/stable/resrep42765.

[4] L. Y. Connolly, D. S. Wall, The rise of crypto-ransomware in a changing cybercrime landscape: Taxonomising countermeasures, Computers & Security 87 (2019) 101568.

[5] P. H. Meland, Y. F. F. Bayoumy, G. Sindre, The ransomware-as-a-service economy within the darknet, Computers & Security 92 (2020) 101762.

[6] A. Alqahtani, F. T. Sheldon, A survey of crypto ransomware attack detection methodologies: an evolving outlook, Sensors 22 (2022) 1837.

[7] H. Yousaf, Investigating transactions in cryptocurrencies, arXiv preprint arXiv:2203.14684 (2022).
[8] C. Whelan, D. Bright, J. Martin, Reconceptualising organised (cyber) crime: The case of ransomware, Journal of Criminology (2023) 26338076231199793.
[9] M. A. Harlev, H. Sun Yin, K. C. Langenheldt, R. Mukkamala, R. Vatrapu, Breaking bad: De-anonymising entity types on the bitcoin blockchain using supervised machine learning, in: $51^{st}$ Hawaii International Conference on System Sciences, 2018.
[10] M. Bartoletti, B. Pes, S. Serusi, Data mining for detecting bitcoin ponzi schemes, in: 2018 Crypto Valley Conference on Blockchain Technology (CVCBT), IEEE, 2018, pp. 75–84.
[11] D. Y. Huang, M. M. Aliapoulios, V. G. Li, L. Invernizzi, E. Bursztein, K. McRoberts, J. Levin, K. Levchenko, A. C. Snoeren, D. McCoy, Tracking ransomware end-to-end, in: 2018 IEEE Symposium on Security and Privacy (SP), IEEE, 2018, pp. 618–631.
[12] H. H. Sun Yin, K. Langenheldt, M. Harlev, R. R. Mukkamala, R. Vatrapu, Regulating cryptocurrencies: a supervised machine learning approach to de-anonymizing the bitcoin blockchain, Journal of Management Information Systems 36 (2019) 37–73.
[13] C. G. Akcora, Y. Li, Y. R. Gel, M. Kantarcioglu, Bitcoinheist: Topological data analysis for ransomware detection on the bitcoin blockchain, 2019. arXiv:1906.07852.
[14] G. Kappos, H. Yousaf, R. Stutz, S. Rollet, B. Haslhofer, S. Meiklejohn, How to peel a million: Validating and expanding bitcoin clusters, in: 31st USENIX Security Symposium (USENIX Security 22), 2022, pp. 2207–2223.
[15] P. Nerurkar, Illegal activity detection on bitcoin transaction using deep learning, Soft Computing 27 (2023) 5503–5520.
[16] L. Yu, F. Zhang, J. Ma, L. Yang, Y. Yang, W. Jia, Who are the money launderers? money laundering detection on blockchain via mutual learning-based graph neural network, in: 2023 International Joint Conference on Neural Networks (IJCNN), IEEE, 2023, pp. 1–8.
[17] N. Pocher, M. Zichichi, F. Merizzi, M. Z. Shafiq, S. Ferretti, Detecting anomalous cryptocurrency transactions: An aml/cft application of machine learning-based forensics, Electronic Markets 33 (2023) 37.
[18] B. Ampel, K. Otto, S. Samtani, H. Chen, Disrupting ransomware actors on the bitcoin blockchain: A graph embedding approach, in: 2023 IEEE International Conference on Intelligence and Security Informatics (ISI), IEEE, 2023, pp. 1–6.
[19] N. Capuano, G. Fenza, V. Loia, C. Stanzione, Explainable artificial intelligence in cybersecurity: A survey, IEEE Access 10 (2022) 93575–93600.
[20] K. Rozier, Linear temporal logic symbolic model checking, Computer Science Review 5 (2011) 163–203. doi:10.1016/j.cosrev.2010.06.002.
[21] L. W. Cong, C. R. Harvey, D. Rabetti, Z.-Y. Wu, An anatomy of crypto-enabled cybercrimes, Technical Report, National Bureau of Economic Research, 2023.
[22] M. Nazzari, From payday to payoff: Exploring the money laundering strategies of cybercriminals, Trends in Organized Crime (2023) 1–18.