Privacy and Cryptocurrencies - A Systematic Literature Review

Herskind, Lasse; Katsikouli, Panagiota; Dragoni, Nicola

Published in:
IEEE Access

Link to article, DOI:
10.1109/ACCESS.2020.2980950

Publication date:
2020

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Privacy and Cryptocurrencies—A Systematic Literature Review

LASSE HERSKIND, PANAGIOTA KATSIKOULI, AND NICOLA DRAGONI

DTU Compute, Technical University of Denmark, 2800 Lyngby, Denmark

Corresponding author: Panagiota Katsikouli (panka@dtu.dk)

This work was supported by the Danish Industry Foundation (Industriens Fonds) under Grant 2019-0046.

ABSTRACT
Our transaction history in the current centralized banking system has the ability to reveal a lot of private information for each spender, both to the banking system itself, but also to those entities that surround it (e.g., governments, industry etc). Examples of leaking information constitute the amounts spent, the goods on which the amounts were spent, the spending locations and the users we exchange money with. This knowledge is powerful in the hands of those who have it, and can be used in multiple ways, not always to our benefit. Cryptocurrencies, such as the famous Bitcoin, were proposed as a means to address the limitations of centralized banking systems and to offer its users privacy with regards to their transactional data. In this work, we perform a systematic literature review on the realm of privacy for electronic currencies. We present the development of digital money from electronic cash to cryptocurrencies and focus on the techniques that are employed to enhance user-privacy. Furthermore, we present flaws of the current cryptocurrency systems, which reduce the privacy of the cryptocurrency users. Finally, we describe three research directions to enhance privacy for cryptocurrencies: transaction propagation mechanisms, succinct ZK proof systems without a trusted setup, and specialised trustless zero-knowledge proofs.

INDEX TERMS
Anonymity, bitcoin, confidentiality, cryptocurrencies, electronic cash, privacy, zero-knowledge.

I. INTRODUCTION
Data leakage and other privacy related matters receive more and more attention since 2000 and reasonably concern the general public; take for instance the case of Cambridge Analytica [1] or the incident of disclosing medical data over paging systems in Vancouver.¹

A window into our private lives has been opened even before the social media and smart-phone era, with the adoption and use of credit cards for massive and easy spending. Convenience, however, has come at a price, as never before have our spending habits been monitored as much as they are through the credit card system [2]. What is more, the high number of terror-related incidents around the globe, has given rise to a belief that governments with complete knowledge on citizens will be able to enhance protection; as a result, there exists an increased demand for laws² that would permit governments to access citizens’ complete data collected by banks, mobile phone operators etc. Although a government backdoor is not difficult to implement, it is hard to guard against attackers. Even when assuming “impenetrable” security, such backdoors leave open the worrisome door for authorities to decide on the fate of every citizen.

With the financial crisis of 2008, a pseudonymous actor, Satoshi Nakamoto, published a proposal for a new monetary system called Bitcoin [3]. The system provides a purely digital currency that is managed by a network of unknown and untrusted nodes, with no need for a trusted third party. The lack of a trusted party makes Bitcoin significantly different from preceding monetary systems, as control of funds is entirely in the users’ hands. This feature had also led the public to believe that Bitcoin was completely private, and used only by criminals. Ever since, such assumptions are weakening, as research on the currency has shown the traceability characteristic of Bitcoin, and companies such as Elliptic, and Chainalysis provide tracking services for the FBI and CIA.³ While this may restore the cryptocurrencies'
reputation in the public mind, it reveals that Bitcoin does not solve the issue of privacy; if the authorities can track it, so can everybody else.

While the current centralized economic systems offer some level of protection from financial crimes, private transactions are only feasible with the use of cash. Inspired by the decentralized features of cryptocurrencies and the promising steps towards privacy in the field, there is hope that this new monetary system may be able to also support guaranteed privacy and protection against financial fraud.

A. CONTRIBUTION OF THE PAPER
In this paper, we review the existing literature regarding privacy-offering techniques and privacy-related limitations of current electronic currencies. We present a short overview on privacy related aspects of digital cash systems which have become applicable to the more recent cryptocurrency systems.

Similar to a recent comprehensive survey on the topic by Kus Khalilov and Levi [4], we discuss privacy issues and weaknesses of Bitcoin-like systems, as well as their proposed extensions. Unlike previous surveys ([4]–[6]), we conduct a systematic literature review, where (i) we put emphasis on the methodology followed for the mining of the reviewed material and (ii) the mined literature is driven by the posed research questions and the review’s scope.

We provide a taxonomy of the literature in three axes: (a) privacy attacks using information from transaction graphs, (b) privacy attacks using information from the peer-to-peer network, and (c) other unclassified privacy attacks, including cookies and cross-chain inter-services exchange. In each of these groups, we discuss techniques and approaches proposed to limit the information leakage of the attacks. Finally, we identify and discuss three key research directions that could lead to an improvement in anonymity and mitigate shortcomings of current systems, namely: transaction propagation mechanisms, succinct ZK proof systems without a trusted setup, and specialised trustless zero-knowledge proofs.

B. PAPER OUTLINE
The rest of the paper is structured as follows. In Section II we describe the methodology followed for the literature collection. In Section III we discuss the background of electronic cash systems and cryptocurrencies that we deem necessary for the completeness of the study. In Section IV we present and review the flaws, as well as the attacks on the privacy of cryptocurrencies, and present some currently used mitigation methods. In Section V we discuss promising future research directions before concluding the paper in Section VI.

II. LITERATURE REVIEW METHODOLOGY
In this section, we present our methodology for discovering relevant literature. In order to conduct a systematic literature review, we followed standard practices based on the works of Petersen et al. [7] and Wohlin et al. [8].

To start our quest, we define privacy in cryptocurrencies as the ability to perform private transactions. A private transaction has two distinct properties: i) confidentiality, i.e., hiding the amounts, and ii) anonymity, i.e., hiding the sender and receiver [9]. This definition will assist us in the methodology followed for selecting the review material.

A. RESEARCH QUESTIONS
To distil research questions and search queries, as well as to shape the scope of the review, we use the following PICOC criteria [10]:

- **Population**: the set of digital currencies / electronic cash / cryptocurrencies
- **Intervention**: the techniques used to perform private transactions
- **Comparison**: we compare the techniques by their compatibility with existing currencies, their objective and how well they protect the privacy of the users
- **Outcomes**: we present privacy-techniques, their limitations and a direction for future research within this area
- **Context**: the comparison will be performed within academia, i.e., by scientific papers analysing the protocols

Based on this method, we pose the following research questions:

1) What are the existing protocols and techniques for enforcing privacy in current cryptocurrencies?
2) How well do these protocols and techniques enforce privacy, and how can they be attacked?
3) How can we improve privacy in cryptocurrencies?

1) SEARCH QUERIES
To answer these research questions, we construct a set of keywords and key-phrases that can help us discover a wide range of the existing systems and their weaknesses. Our set of search-queries is: **anonymous electronic cash, anonymity attack digital cash, anonymous digital cash, anonymity attack digital cash, anonymous cryptocurrency, anonymity attack cryptocurrency.**

B. SELECTION METHODOLOGY OF REVIEWED MATERIAL
Our search methodology follows the sequence of stages proposed in [7]:

1) Initial retrieval of the result from the search-queries
2) (Automated) Duplicate removal, in order to include the most recent version of a paper
3) (Automated) Perform inclusion and exclusion based on title and abstract
4) Perform manual duplicate removal
5) Perform manual inclusion and exclusion based on titles and abstracts
6) Full-text read-through
7) Snowball sampling

As proposed in Petersen et al. [7], we design and apply a number of exclusion and inclusion criteria (EC and IC,
respectively) to retrieve the subset of published material that best suits the scope of our review:

- **EC1**: paper is not accessible in full-text
- **EC2**: paper is not presented in English
- **EC3**: the paper has no title or abstract
- **EC4**: the proposal of the paper requires additional hardware (e.g., a physical e-wallet)
- **EC5**: the paper has less than \( \min((2019 - \text{year}) \times 2, 5) \) citations
- **IC1**: the work should use, analyse or propose an anonymity-enhancing protocol
- **IC2**: the work should use, analyse or propose an electronic cash system
- **IC3**: the work should use, analyse or propose a decentralised protocol in the field of digital currencies

A paper is excluded if it satisfies any of the EC’s and accepted only if it satisfies IC1 together with either IC2 or IC3 and no EC’s. We use this construction as relevant protocols may come from outside the cryptocurrency space. An example is anonymous broadcasting, which does not satisfy IC2 but is useful for privacy-preserving currencies. To enforce the IC’s, we construct a set of keywords; for a paper to be deemed relevant, we set a threshold of at least 2 keywords to be included in the title or abstract of the paper.

**Keywords:** peer to peer, p2p, peer-to-peer, privacy, private, ledger, weakness, vulnerability, vulnerable, attack, challenge, de-anonymising, trace.

In relation to IC3, the keywords peer to peer and ledger are necessary. Following IC1, we focus on the area of private currencies, thereby including privacy and private. Furthermore, we include keywords related to weaknesses, attacks and challenges of current protocols, as well as papers looking into de-anonymising and tracing transactions, in order to identify suitable research directions.

At this point, we would like to highlight the evolution of the key topic in the mined literature, with respect to the above-mentioned keywords. After identifying the 10 most popular keywords among the papers in our mined collection, we grouped those in three clusters, privacy, blockchain and vulnerable. Privacy includes keywords regarding anonymity and non-traceability, blockchain includes keywords regarding the ledger technology and its applications and finally, vulnerable includes keywords regarding attacks and weaknesses.

**C. SCREENING PROCESS**

In this section, we elaborate upon the actual process of finding relevant papers. We show in Fig 2 the number of relevant papers published each year. The various stages, as well as the number of papers found and excluded at each stage throughout the process, are shown in Figure 3.

Our primary search was performed using the DTU Find-it database service, which taps into ACM, IEEE, Scopus, CiteSeer, arXiv and other widely used journals and databases. The search was performed on the 15 of August 2019.

1) **INCLUSION AND EXCLUSION BASED ON TITLE AND ABSTRACT**

By applying the keywords used for exclusion and inclusion, we removed a number of papers, the majority of which do not investigate the anonymity properties, but merely state that some currency is anonymous.
2) MANUAL DUPLICATE REMOVAL
This stage is useful for cases of short-papers and poster-papers that are found as duplicates of the full conference papers already mined by our search.

3) MANUAL EXCLUSION ON TITLES, ABSTRACT AND AVAILABILITY
From the manual process, we found that some papers focus on specific use-cases, e.g. energy-trading, utilising cryptocurrencies as a means of payment, or a peer-to-peer blockchain for some service. As these papers are not relevant to our review, we applied our exclusion and inclusion criteria and discarded some of them based on the title, abstract or non full-text availability.

4) MANUAL ADDITIONS
Beyond papers found via our search method, we have included 2 relevant papers ([11], [12]) based on presentations at the conference Theory and Practice of Blockchains 2019 at Aarhus University.

5) SNOWBALL SAMPLING
We perform forward and backward snowball sampling as described by Wohlin [8], to account for vital papers, that may not have been included in our initial search because they were not caught by our queries or may have been published outside the reach of DTU Find-it. We select potentially relevant papers based on where they have been cited, their title, and the authors.

III. BACKGROUND OF DIGITAL MONEY
In this section, we make a short passage through the history of digital money, on aspects relevant to our review. In particular, we present a variety of electronic cash systems, in order to highlight the various techniques, such as blind signatures, used to provide such systems with anonymity. We focus on these techniques as they have become relevant and are also used more recently in cryptocurrencies. We also provide an introduction to cryptocurrencies, and specifically to Bitcoin, where the key terminology and system actions are presented.

A. A PRIMER ON ELECTRONIC CASH
The idea of transferring electronic cash supported by a bank, without the bank having knowledge on specific details, is credited to David Chaum, who in 1983 proposed the idea of an anonymous electronic cash system that utilises blinded signatures [13]. A blinded signature scheme allows an entity to sign a message, without exposing its contents. This is done by first concealing the message, and then signing the concealed message; the concealed signature can then be verified publicly similar to regular digital signatures.

A practical implementation of the scheme takes the form of digital notes, with a fixed monetary value of 1 USD. The scheme then utilises blindly signed notes to ensure the anonymity of the user unconditionally; but only as long as the user is honest. To ensure that a user is unable to double-spend their notes, the scheme splits every note (1 USD) into a set of “subcomponents”, of which only half is transferred during payment.

A limitation of the system is that coins are non-transferable. For example, after a transfer from user A to user B, the coins have to be redeemed at the bank and cannot be passed to a third user. It worth to note that it is only during withdrawals that double-spending can be spotted.

Addressing the issue of coin transferability, Hayes [14] proposed a system in which the current owner of the coin signs the chain of transactions; i.e., when C receives the coin, a chain of transactions partly signed by A and B is contained in it. To mitigate leakage of the owners identities in these transaction-chains, the scheme proposes the use of pseudonyms, of which the user can generate arbitrarily many [14].

A number of Electronic Cash systems with varying revocation capabilities are proposed in [15]. As noted by the authors, the ability to revoke anonymity only in cases of double-spending, facilitates criminals to perform other perfect crimes. An example of this is money-laundering; as no double-spending occurs, the identity of the launderer remains perfectly hidden. From this, the term “fair cash system” was coined. A fair cash system is a monetary system where an honest user is to be protected, but a malicious user should be deanonymized. However, as malicious behaviour is not always as well defined as double-spending is, such solutions require the introduction of some trustee; for example, an honest and righteous judge. Hou and Tan [16], [17] propose to use group-signatures and a judge as manager to perform auditable tracing, i.e., support detection of illegal tracing. Similarly, Pfitzmann and Sadeghi [18] proposes to let the user act as a trustee for deanonymizing malicious entities.

Decentralised use of electronic cash systems was firstly introduced by Belenky et al. [19] and Figueiredo et al. [20]. In these works, the use of an anonymous electronic cash system is proposed to incentivise participants in Peer-2-Peer (P2P) networks and to limit freeloaders. Palaka et al. [21] suggested the use of a P2P network to transact coins while the bank is still participating in the minting of funds. In order to address double-spending, from which those systems were suffering, Osipkov et al. [22] proposed the use of a Distributed Hash Table for keeping track of spent coins. While this approach decreases the power and necessity of the bank at transfer-time, their solution requires trusted witnesses and only supports probabilistic searching.

B. THE BIRTH OF A CRYPTOCURRENCY - BITCOIN
In the Bitcoin paper [3], Satoshi Nakamoto combines previously tested ideas [14], [22] into a fully-fledged currency that exists without the cooperation of a bank, or other entity of authority. Bitcoin utilises a ledger of blocks to record every transaction that has happened in the network, and thereby provides a shared “truth” between the network peers. A block consists of a set of transactions, a hash reference...
to the previous block and a nonce used in the transaction verification mechanism (Proof-of-Work).

A transaction in Bitcoin is a transfer of some value between pseudonyms known as addresses (hashes of public keys). By owning the matching private-key, one can transfer the Bitcoin that the address owns. As Bitcoin is based on unspent transaction outputs (also called UTXO), and not a balance, the address will in practice control a series of outputs containing the funds. For some output $O$, we write $val(O)$ for its value and $owner(O)$ for its owner.

A transaction consists of a series of sources, destinations and signatures to prove ownership of the sources. A source is a previous transaction output (or coinbase/block reward) that has not yet been spent, and a destination is a freshly generated output. For ease of notion and to distinguish between sources and destinations, we denote $s_i$ for a source $i$ and $d_j$ for a destination $j$. We then define a transaction message $M$ with $n$ sources and $m$ destinations as the tuple:

$$M \overset{\text{def}}{=} (s_1, \ldots, s_n; d_1, \ldots, d_m) \quad (1)$$

A valid transaction (denoted by $tx$) is then a tuple of a message $M$ and of signatures by every source-owner, where $M$ satisfies that no value is lost or created. This is enforced by requiring that every source $s_i$ is unspent, that all destination-values $val(d_j)$ are non-negative, and that the sum of source-values is equal to the sum of destination-values.

Let $\sigma_{owner(s_i)}(M)$ be a signature on the message $M$ by the owner of $s_i$. The transaction

$$tx \overset{\text{def}}{=} (M, \sigma_{owner(s_1)}(M), \ldots, \sigma_{owner(s_n)}(M)) \quad (2)$$

is valid subject to

$$\sum_{i=1}^{n} val(s_i) = \sum_{j=1}^{m} val(d_j), \quad \forall j (val(d_j) \geq 0) \quad (3)$$

When users wish to transfer only part of a source, they can perform transactions with an additional destination to an address controlled by themselves (known as change). For example; let Alice be a user with a source of 50 Bitcoins who wishes to transfer 10 Bitcoins to Bob. She, therefore, transfers 10 Bitcoins to Bob and 40 Bitcoins to herself as change.

By storing every transaction on the shared ledger, every peer can identify the valid transactions and reject those that would double-spend or be invalid.

To add transactions on the shared ledger, the Bitcoin network performs a collective run of a sybil-resistant lottery mechanism, the Proof-of-Work. In practice, this is achieved by creating a block of unconfirmed transactions and updating the nonce until the hash of the created block satisfies the difficulty criteria, i.e., until the hash has a minimum number of leading zeros. The peer creating the first block that satisfies this, is the "winner" who may publish the block and receive a block reward. This mechanism is known as the "consensus mechanism", and allows any peer to use some scarce resource, here computation, to increase its probability of winning the lottery [23]. Chaining the blocks ensures that to temper with a block would require as much work as the network has already done since the creation of that block, thereby providing "computational" integrity.

The idea of incentivising participation through monetary rewards has also been explored in [19], [20]. However, the reward has previously been unrelated to the purpose of the system, limiting the systems ability to be self-contained. To the best of our knowledge, Bitcoin is the first network where the value of the reward is supplied by the participation of peers. In other words, the currency has value because it is tamper-proof and it is tamper-proof because it has value. As noted by Fischer et al. [24], “real” consensus protocols have issues of non-termination. That problem is known as availability. Consequently, such consensus protocols may not be particularly useful for the case of monetary transactions, where the availability of the system is of high importance. Due to this, the notion of consensus used within Bitcoin is probabilistic. This means that there is a chance that history is reverted; however, the nature of the blockchain ensures that the likelihood decreases exponentially with the age of the block; that is, by adding several blocks after a particular block, the probability of reverting it becomes negligible.

IV. BITCOIN IS NOT PRIVATE

In this section, we review the literature we mined using the methodology described in Section II, which highlights significant flaws of the Bitcoin cryptocurrency system regarding the anonymity and privacy of the users and of the transactions between them. Our goal is to challenge the notion given by the media and the public regarding the privacy of cryptocurrencies [4]–[6], [25], [26], and in particular of Bitcoin, to highlight where the limitations are coming from, but also to present directions for improving the weak privacy guarantees.

We recall that by privacy in cryptocurrencies, we refer to the ability to perform transactions while satisfying two distinct properties: i) confidentiality, i.e., hiding the transaction amounts, and ii) anonymity, i.e., hiding the sender and receiver [9].

The section is structured in three parts, each related to an attack vector: Transaction Graph Analysis, Peer-2-Peer Network Analysis and other, unclassified deanonymization approaches. In each case, we introduce the context of the attack vector and review deanonymization techniques as well as proposed approaches for improving the anonymity guarantees.

A. TRANSACTION GRAPH ANALYSIS

Here, we review various methods that analyse the transaction graph of cryptocurrencies in order to identify users. A transaction graph is one where vertices correspond to transactions and edges to flows of funds. Let $G$ represent such a graph, an example of which is shown on the left-hand side of Figure 4. From $G$ we can construct the address graph $H$,
where vertices correspond to addresses and edges represent the flows of funds between these addresses (see the right-hand side of Figure 4). Based on $H$ an attacker can cluster the addresses to infer users by using simple heuristics, such as the multi-source-same-owner heuristic [27], and generate the graph $J$, where vertices are such clusters and edges show the flows of funds as before; that is depicted with the shaded area on the right-hand side image of Figure 4. Given the graph $J$, an attacker can link transactions to a particular user as well as reveal relationships between users. This information can be used for user identification, and to reveal the complete transactional history of the user.

In the following discussion, we consider the transaction graph in the style of Bitcoin, as described above (note that a balance-based system as Ethereum is similar to the address graph). Most works that use the Bitcoin transaction graph are mainly applying pattern recognition and clustering techniques [6], [28], [29]; as such, in this section, we will focus on works of this category.

In early work by Androulaki et al. [29], the authors consider the setting where Bitcoin is used in a university by students, workers and professors, for everyday purchases (i.e., food & books). They show that by using simple heuristics, a considerable fraction of the users (approximately 40%), could be deanonymized with high accuracy (80%). Because shops have a fixed pseudonym (address), the tuple (location, address), along with the amounts spent by users at a shop, can be used to fingerprint the behaviour of users. The authors note that users seldom perform transactions with more than two destinations (one returned as the change to themselves and another for the actual payment at the shop) due to the implementation of wallets. Furthermore, the authors show that having a large fraction of users generating multiple new addresses, does not significantly decrease the attacker’s ability to identify users, when compared to random guesses. In addition, Meiklejohn et al. [27] show that the clustering of addresses can enable an attacker to identify users. They note that collusion with a large exchange, performing KYC (Know Your Customer), trivialises linkage between clusters and identities. Building on the works of Meiklejohn et al. [27] and Androulaki et al. [29], Fleder et al. [30] show that transaction graph analysis using known addresses in Bitcoin, reveals information that can be used to make predictions on the users and their addresses. They note that, scraping Bitcoin forum-tags allows direct linkage between addresses and identities which, together with the clustered transaction graph, reveals the user’s transaction history.

In [31] Meiklejohn and Orlandi note that unlinkability in a Bitcoin setting is not possible, as coins contain their history. The authors propose a new measure of anonymity, namely taint resistance, which measures an attacker’s inability to link a transaction’s source with its destinations, given the source is tainted. To improve this, they propose a practical implementation of the CoinJoin scheme envisioned by Maxwell [33].

More recent works have focused on using the findings of earlier works (such as [27]–[31]) for law enforcement purposes. In [34], a graphical tool is proposed to make

---

5An adversary’s inability to link a coin to the transaction spending it [32].

6CoinJoin is a distributed method for combining multiple payments into a single transaction.

7For example, for tracking purchases of illegal substances from known dealers and money laundering, https://www.chainalysis.com/, accessed 10-11-2019
transaction graph analysis easier for non-experts. Furthermore, Chen [35] has looked into how crypto markets transact, showing that large mining pools and markets utilise peeling chains and off-chain databases to act as inbuilt mixers (we discuss those further in the manuscript).

An overview of these methods is presented in Table 1.

### TABLE 1. Overview of transaction graph analysis techniques, commonly used to analyse the Bitcoin blockchain.

<table>
<thead>
<tr>
<th>Objective</th>
<th>Method</th>
<th>Requirements (data)</th>
<th>Papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clustering addresses</td>
<td>Follow</td>
<td>Transaction Graph</td>
<td>[27],</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Bitcoin blockchain)</td>
<td>[28]</td>
</tr>
<tr>
<td>Recognize user type</td>
<td>Pattern recognition</td>
<td>Transaction Graph &amp;</td>
<td>[29]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Known addresses</td>
<td></td>
</tr>
<tr>
<td>Link clusters to users</td>
<td>Scrape addresses from</td>
<td>Transaction Graph &amp;</td>
<td>[30]</td>
</tr>
<tr>
<td></td>
<td>website</td>
<td>Known addresses</td>
<td></td>
</tr>
</tbody>
</table>

B. APPROACHES FOR LIMITING INFORMATION LEAKAGE FROM THE TRANSACTION GRAPH

As shown in the previous paragraphs, information extracted from the analysis of the transaction graph poses a significant threat to the users of Bitcoin and similar systems. Here we review schemes proposed to limit this threat. While these schemes have the same goal, i.e., to improve privacy, the methods employed and their compatibility with existing cryptocurrencies vary significantly. We split the schemes into two distinct groups based on their compatibility, or not, with existing cryptocurrencies:

- **Compatible schemes**, which can be incorporated as extensions in existing cryptocurrencies (Bitcoin & Ethereum) with no or minimal changes to the underlying protocol.
- **Incompatible schemes**, which use a radically different protocol or require an extensive update to the protocol and sometimes to the security assumptions.

#### 1) COMPATIBLE SCHEMES TO EXISTING Cryptocurrencies

We review two schemes in this category: **CoinJoins** and **Zero-Knowledge proofs**.

**a: COINJOINS**

A similarity between the extensions to current cryptocurrencies is that they support mixing of coins, called *coinjoins*, thereby improving *taint resistance* [31].

The idea behind *coinjoins* is rather intuitive. Given multiple transaction messages (defined per Equation 1), combine the sources and the destinations of the messages to create one large transaction message *M* as seen in Equation 4.

\[
M = (s_1, \ldots, s_{n_1+n_2}; d_1, \ldots, d_{n_1+n_2})
\]

The transaction is then completed in the same manner as described in Equation 2. A visualization of this idea is shown in Figure 5, where the *Mixer* can be a centralised server or a decentralised protocol. Furthermore, the idea of mixing does not require any new primitives; therefore, it can be applied to the Bitcoin ecosystem without any changes [4].

The scheme was formally described as **CoinJoin** by Maxwell in [33]. At its inception, the mixing was done through a centralised service, which had to be trusted to protect the privacy of the users and to not steal coins [26]. While these services could provide decent mixing, as Meiklejohn noted in [27], some mixers built unmixed transactions, while others stole their coins. What is more, such a scheme would often require every transaction included in the mix to be of the same value, in order to make linkage non-trivial. In [36], a practical implementation of a decentralised coinjoin protocol was proposed, called *coinshuffle*. Ruffing et al. [37] improved upon the efficiency of coinshuffle, utilising a mixnet to create *coinshuffle++/dicemix*. Extending on the CoinJoin scheme, Maurer [38] proposed a method to perform CoinJoins with arbitrary values.

Taking a different approach, Ziegeldorf et al. [39] proposes swapping destinations instead of joining transactions, thereby removing the link element entirely. To illustrate the point, let \( A \rightarrow B \) be a transaction from user \( A \) to user \( B \). Assuming we have two transactions \( A \rightarrow B \), and \( C \rightarrow D \) of equal values, we create the transactions \( A \rightarrow D \) and \( C \rightarrow B \), instead of the CoinJoin \((A, C) \rightarrow (B, D)\). The intuition is that we throw away the true link, rather than hiding it within a set of decoys.

The aforementioned methods of CoinJoin are cooperative and require additional work from the user. Therefore, only few joined transactions are actually performed. To mitigate this issue, Meiklejohn and Mercer [40] proposed the *Möbius* scheme, which uses a smart-contract on the Ethereum network to improve taint resistance while decreasing the required cooperation between participants.

**b: ZERO-KNOWLEDGE PROOFS**

Simply put, a Zero-Knowledge proof is a proof that convinces a verifier of some statement, without revealing any information other than that the statement is true [41]. Within cryptocurrencies, this is useful for privacy, as it allows a user to prove that she has sufficient funds for some transfer, 8https://www.longhash.com/news/coinjoins-as-a-percentage-of-all-bitcoin-payments-have-tripled-to-409-over-the-past-year, accessed 22-09-2019
TABLE 2. Overview of compatible approaches to limit information revealed from transaction graph analysis in Bitcoin and Ethereum.

<table>
<thead>
<tr>
<th>Proposal</th>
<th>Method</th>
<th>Advantages</th>
<th>Disadvantages</th>
<th>Papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centralised Mixer</td>
<td>Server receives coins, perform mixing and publish a mixed transaction to the network.</td>
<td>Easy to do, can enforce large anonymity-set.</td>
<td>Requires trust in the executing server.</td>
<td>[26],  [27]</td>
</tr>
<tr>
<td>Decentralised Mixer</td>
<td>A group cooperate and mix their transaction to one larger transaction.</td>
<td>No trusted server.</td>
<td>Possible set of sources. Vulnerable to DoS and Sybil attacks.</td>
<td>[36]- [38], [40]- [42], [45]</td>
</tr>
<tr>
<td>Coin Swapping</td>
<td>A group of participants conduct atomic transfer to the others recipients.</td>
<td>No obvious linking.</td>
<td>Require trust in the &quot;sender&quot;, as he can leak the true transaction.</td>
<td>[39]</td>
</tr>
<tr>
<td>BitFlow</td>
<td>Zero-Knowledge range proofs prove non-negative value transfers.</td>
<td>Leaks no knowledge of the transaction value.</td>
<td>Did not hide the transaction graph.</td>
<td>[42]</td>
</tr>
<tr>
<td>Zether</td>
<td>ZK range proofs and ring signatures.</td>
<td>Leaks no knowledge of the transacted value.</td>
<td>Possible set of sources. Expensive computation.</td>
<td>[12]</td>
</tr>
</tbody>
</table>

without disclosing the amount of her funds, the amount she transfers or which output is used as a source.

Most of the zero-knowledge proof schemes have been implemented in distinct currencies; due to this, in this work we will focus on works that have used the concept as smart contract implementations in Ethereum.

In [42], Herskind et al. show how Zero-Knowledge range proofs, as proposed by Bünz et al. [43], could be used to support confidential transactions between two "known" senders, without disclosing the transferred amount. In this work, the knowledge of the attacker is limited through concealment of the transaction values. Independently from [42], Bünz et al. [12], at the same time, proposed Zether, a scheme that allows confidential and anonymous transactions on the Ethereum network by using ring-signatures [44] on top of the range proofs.

A summary of the compatible approaches to limit the information revealed from the analysis of the transaction graph is offered in Table 2.

2) INCOMPATIBLE SCHEMES TO EXISTING CRYPTOCURRENCIES

While most of the schemes that are incompatible with the most popular existing cryptocurrencies introduce new privacy methods (e.g., stealth addresses, confidential transactions), some extend upon the idea of mixing.

In particular, the idea of using a non-interactively aggregatable signature scheme to perform CoinJoins is proposed in [46]. Not only would it support easy aggregation of transactions between users, it would also allow a miner to aggregate every transaction within a block, resulting in each block being one large CoinJoin transaction.

Methods that aim at mitigating the Bitcoin traceability issues, have been implemented by building fully-fledged cryptocurrencies. Perhaps the most popular among these are the Monero and the Zcash currencies, discussed in the following paragraphs.

3) MONERO

To enhance the anonymity of users, Monero [47] applies ring-signatures [44] and a set of possible sources (rather than conclusive sources and signatures), as seen earlier for Bitcoin. This means that there is no explicit notation of who performed the transfer, only that it was one of the users in the ring. On top of this, a stealth address (i.e., a one-time address related to some private-key [5]) is used as the receiving address.

In the first version of the Monero scheme (i.e., pre-RingCT update) confidential transactions were not enforced, making it hard for users to create a ring of transactions with the same value. As a result, rings often had a size of 1 (i.e., including only the actual spender), making it trivial to follow the "private" transaction [48], [49]. As noted in [5], the use of a ring-signature provides limited privacy, as the anonymity-set is often limited [49]. In updated versions, the transacted values are hidden by confidential transactions, making it easier to create a useful ring, since any confidential transaction can be used [50]. However, as shown by Moumlser et al. [49], the method of picking mixins (i.e., decoy sources) poses a significant threat to the anonymity of the users, and the actual source could usually be detected by using just the age of sources.

While the improvement of confidential transactions and the introduction of new mixing-choosing mechanisms significantly improved the anonymity of Monero users [49], attacks on the ring-signature and Unencrypted Payment ID [51], [52] still pose a privacy threat. In addition, a careless participant could use multiple outputs from the same transaction and thereby break Monero’s unlinkability9 [48].

4) ZEROCOIN

The Zerocoin project, proposed by Miers et al. [32], presents a scheme that allows a user to deposit a Bitcoin coin into a pool of "Zerocoins". The Zero-knowledge protocol uses commitments to Bitcoins and serial numbers to perform a non-interactive proof and withdraw funds from the pool. The proposal has significant reservations and a proof size of 25 KB (basic Bitcoin transactions are approximately 250 bytes). While this was improved to 10 KB by reducing soundness of proofs [53], the Zerocoin transaction size is still substantial in comparison to a typical Bitcoin transaction.

5) ZEROCASH

In the wake of Zerocoin, a new scheme, called Zerocash, which uses a relatively new type of cryptography, Zero-Knowledge Succinct Argument of Knowledge proofs (ZK-SNArKs), was proposed. The scheme significantly improves on some of the drawbacks of Zerocoin,
by greatly reducing the size of proofs from multiple KB to 288 Bytes [54], [55]. This is achieved with the use of a trusted setup to support succinct proof constructions [32], [55], [56]. In order to explain the idea behind shielded transactions in Zerocash, let us consider two sets of coins, one denoting every coin that ever existed, and one denoting commitments to spent coins. By using ZK-SNARKs, Zerocash can prove that the sources of a new transaction are within the set of existing coins but have not been spent before, without disclosing the sources themselves. This means that Zerocash, similar to Monero, keeps an ever-expanding set of “coins” and “spent coins” [11]. In practice, the Zerocash implementation Zcash, uses multiple sets of coins, a transparent set as in the Bitcoin system, and a shielded-pool where transactions employ ZK-SNARKs to prove validity while hiding the specifics. To address the regulatory complications, an addition supporting taxation of shielded transactions is proposed in [57].

The shielded transactions are shown to be fully anonymous with strong security guarantees [58]. However, the use of the shielded pool is minimal (1.3% of transactions in September 2019) [10] with users impeding anonymity through their leaking behaviour (deanonymization of 69.1% of the shielded transactions) [58]. Another issue faced in Zcash is that large fractions of the network can be deanonymized; [59] shows that transactions can be linked to IP-addresses with 50% precision and 82% recall. To address this issue, Kappos et al. [60] propose the use of Mixnets for anonymous broadcasting.

6) MIMBLEWIMBLE

Initially conceptualized and introduced by the pseudonymous individual “Tom Elvis Jedusor” at a Bitcoin IRC channel, MimbleWimble was refined and soundly presented in [61] by Andrew Poelstra.

MimbleWimble eliminates the idea of performing transactions between addresses. This is achieved with the use of a binding and hiding commitment scheme [62], together with range proofs [43]. By using the commitments directly as outputs, the coins act as having their own “private key”, the knowledge of which, along with the knowledge of the value, would enable someone to spend the funds.

In this scheme, a transaction consists of i) sources, ii) destinations (here homomorphic commitments) and iii) a kernel. For simplicity, let us assume that this kernel consists of a signature, a public key (called excess) and range-proofs for the destinations. The excess can be computed by everyone and is derived as the difference between the sum of the source and the sum of the destination commitments. Due to the nature of the commitment scheme [62], this excess will be a valid public key only if no value is generated or lost in the transaction. What is more, the excess’ private key can only be derived by someone that knows each source’s private key. This allows anyone to verify that a transaction does not generate or destroy funds and that only the owner of the sources could have created the transaction by validating the signature and the range-proofs. The range-proofs are used to ensure that no inflation occurs, by proving that the value of every destination is non-negative.

MimbleWimble supports non-interactive CoinJoin transactions by joining transactions, i.e., creating a transaction with the combined set of sources, combined set of destinations, and a set of signatures from the individual transactions. To reduce storage requirements, Poelstra [61] propose a signature scheme that allows non-interactive signature aggregation. Beyond CoinJoin, MimbleWimble supports pruning of intermediate outputs, i.e., cut-through. For example, let us consider an output \( O_k \), which is a destination of \( tx_1 \), and a source of \( tx_y \). Then, the difference \( \sum_i (O_i) - \sum_i (I_i) \) can be computed without knowledge of \( O_k \) and the output can be discarded. An example of the scheme is shown in Figure 6. Because of the non-interactive nature of transactions, block-wide coinjoins are possible, and each block can be seen as a large transaction, supporting pruning across multiple blocks [46], [61].

An introduction [63], as well as an investigation [64] of the possibilities and issues of this scheme have recently become available, exploring the more established signature schemes of Schnorr [65] and BLS [66]. Also, the abilities of the schemes to enable recursively aggregatable signatures are explored in [67]–[69].

While the MimbleWimble scheme provides strong anonymity in the case that an attacker has access only to the blocks (aggregated transactions), an attacker can still act as a network observer, as in [70], [71], and construct the transactions-graph. If an attacker succeeds in that, they can easily afterwards link sources with destinations, allowing them to perform a variety of the attacks, as discussed earlier.

A summary of the schemes, grouped into decoy-based and zero-knowledge categories, is given in Table 3. A more holistic view of methods and currencies (including schemes that are yet to be discussed) is given in Table 7 and Table 8 respectively.

### TABLE 3. An overview of the incompatible schemes that aim at improving privacy and hinder graph analysis. The methods are grouped into two categories: decoy-based and zero-knowledge.

<table>
<thead>
<tr>
<th>Type</th>
<th>Projects</th>
<th>Advantages</th>
<th>Disadvantages</th>
<th>Papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoy-based</td>
<td>Monero and Mimble-Wimble</td>
<td>Relatively easy to understand. Using well known cryptography.</td>
<td>Limited anonymity, hides in a &quot;small&quot; crowd of other transactions.</td>
<td>[47],</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[61]</td>
</tr>
<tr>
<td>Zero-Knowledge</td>
<td>Zerocash and Zerocoin</td>
<td>String anonymity, transactions are hidden within large sets (( &gt; 2^{10} ))</td>
<td>No transaction graph.</td>
<td>[32],</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[53]</td>
</tr>
</tbody>
</table>


### C. P2P NETWORK ANALYSIS

The P2P network in which cryptocurrencies interact, can provide powerful deanonymization tools for an attacker. The P2P network can be modelled as a directed graph \( G \), where
vertices are the peers and edges are the outgoing connections between the peers. Messages are transferred along the edges, and the source is the node originating a message.

Networks have been used to model a large variety of systems in various fields, and as such, multiple techniques are known for identifying the originator of a message or an action in such a network. In the field of disease outbreaks, for example, Pinto et al. [72] show how the network type influences the required number of observer nodes necessary to locate the true source of an event (disease) with high probability. It is found that an attacker capable of placing observers in high-degree nodes, significantly reduces the number of necessary observers.

The idea of observer-nodes in the field of cryptocurrencies was used in [70]. The authors show that anonymity in the Bitcoin network is very limited, and that applying observer-nodes facilitates effective linkage between transactions and the Bitcoin node from which the transactions “originated”, thereby linking addresses with IP-addresses. The authors map 1162 addresses to distinct IP-address, by looking at the relayed packets. Further, they find that monitoring network packets allows them to ignore the use of mixing services.

In the works [4], [73], it is noted that while high-latency networks can provide good anonymity, the popularity of low-latency networks provides a more practical anonymity improvement. As found in [73], an attacker that can trick a user at the application layer to use Java applets, or similar, can bypass the proxy settings.

Biryukov et al. [71], extending on [70], achieves deanonymization of up to 60% of the network, if smaller DoS (Denial-of-Service) attacks are accepted. In the same work, it is argued that an attacker with sufficient power may partition the network and create a relative reality for parts of the network, thereby allowing double-sends to occur. Furthermore, the author highlights that Bitcoin users looking towards the anonymity network Tor in hope of enhanced privacy, are in for a surprise: an attack sustaining a relative reality of the participants using Tor on a network-scale level costs as low as 2500 USD a month.

Beyond the use of observer nodes [70], [71], Fanti and Viswanath [74] notes that an attacker with the power of an ISP is able to deanonymize users by linking the source of transmission directly to a specific IP-address. Furthermore, the authors note that while Bitcoin core developers have updated the mode of propagation from flooding to diffusion, there is no clear evidence that this has improved the resilience of the network to such attacks. In addition, it is shown that attackers who can reach a large number of concurrent corrupt connections to Bitcoin nodes have a high probability of detecting the real source due to the diffusion mechanism.

Other weak points of the P2P network, constitute the network’s actual topology [72], [75] and the use of Simple Payment Verification clients [75] (as users need to trust these to relay “relevant” transactions correctly). As Neudecker and Hartenstein shows in [75], there is little incentive for the participants in the network to forward transactions and blocks, especially in a manner that hides the participants. Therefore, scheme designers may want to consider the following tradeoff: increase DoS resistance and performance or improve anonymity, topology hiding and cost of participation. An example of such tradeoff is the cost of outgoing connections; high costs limit the power of observer-attacks [74] but make relative reality attacks easier to perform. Low costs of outgoing connections have the opposite effect.

A summary of the P2P Network based attack methods and their objectives is shown in Table 4.

D. APPROACHES FOR MINIMIZING P2P NETWORK BASED ATTACKS

Building on [36], Ruffing et al. employ the idea of mixnets, to implement Coinshuffle to address network resistance and performance [37]. However, as noted in [77], Coinshuffle is vulnerable to DoS- and Sybil attacks and does not support confidential transactions, thereby forcing users to find participants with identical mixing values, limiting thus the anonymity set. An extension to the Coinshuffle++ concept supporting confidential transactions is presented in [45].

11That is, anonymous broadcast between n mutually distrusting peers
TABLE 4. Overview of the P2P Network based attack methods.

<table>
<thead>
<tr>
<th>Objective</th>
<th>Domain-specific</th>
<th>Method</th>
<th>Requirements</th>
<th>Papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locating Source in P2P</td>
<td>No</td>
<td>Timing and relaying of received messages.</td>
<td>Curious nodes within the network</td>
<td>[70], [72]</td>
</tr>
<tr>
<td>Locating Source in Tor</td>
<td>No</td>
<td>Traffic analysis of proxy settings or access to a specific server</td>
<td>A Tor server</td>
<td>[73]</td>
</tr>
<tr>
<td>Locating Source of Transaction in Bitcoin</td>
<td>Yes</td>
<td>Timing and relaying of received messages.</td>
<td>Nodes within the network</td>
<td>[71]</td>
</tr>
<tr>
<td>Denial of Service and Locating Source of Transaction in Bitcoin (Tor)</td>
<td>Yes</td>
<td>Relative Reality.</td>
<td>Tor Nodes</td>
<td>[76]</td>
</tr>
<tr>
<td>Locating Source of Transaction</td>
<td>Yes</td>
<td>Utilise ISP information.</td>
<td>Powerful attacker with ability to control the connection of the user.</td>
<td>[74], [75]</td>
</tr>
<tr>
<td>Linking transactions and identities</td>
<td>Yes</td>
<td>&quot;Legging the&quot; transactions that is relayed to light-clients</td>
<td>Verification Server</td>
<td>[75]</td>
</tr>
</tbody>
</table>

Beyond combining mix-nets and coinjoins, which is argued to scale poorly for large networks [77], purely network propagation protocols have been proposed in the literature. For instance, Dandelion [78] splits the diffusion protocol into a propagation mechanism with two stages: stem and fluff. A node in the stem phase relays stem-message to a single peer and otherwise broadcasts, while a fluff node broadcasts any message. The nodes will occasionally flip a coin to decide to change stage. An example of such message propagation is shown in Figure 7. For any node that receives a stem message from a peer, it is not clear whether the peer is the source, or just relaying the message. However, as noted in [79], the protocol makes very optimistic assumptions about the other peers in the network: i) every node knows the complete list of active IP-addresses, ii) every node runs Dandelion, iii) a node creates exactly one transaction and iv) all nodes obey the protocol.

As noted in [79], the Dining Cryptographers network could be used to perform anonymous broadcasting. Briefly, a Dining Cryptographers network [80] works by having a group of peers generate pairwise keys to each other. Every peer will then calculate and publish the sum of their keys modulo 2; the broadcaster inverts the summed key at every index i that is a 1-bit in his message. If a message was transmitted, sum all these summed keys modulo 2 will appear, otherwise, a zero-string. It is immediately clear that such a protocol is vulnerable to DoS attacks, e.g., by multiple peers publishing simultaneously.

The comprehensive survey of Kus Khalilov and Levi [4], shows that the countermeasures available to address privacy issues stemming from analysing the P2P network all build upon Onion routing. A summary of the methods currently explored in the literature is presented in Table 5.

E. UNCLASSIFIED deanonymization APPROACHES

Here, we introduce attack approaches and countermeasures that cannot be included in the previous categories. These approaches include methods that use ad-trackers and exchange API to achieve deanonymization.

1) COOKIES

Goldfeder et al. [81] present an attack method which uses the ad-trackers and cookies applied by most web stores. The authors show that some trackers directly link Bitcoin users (i.e., their personally identifiable information) with the specific transaction on-chain. As noted in the paper, some of these trackers store both unique identifiers (like email addresses or user names) and the actual transaction-ids, making linkage trivial. Interestingly, the authors note that even different websites which use the same tracker, can be used to link individual logs or purchases together and identify both the user and her transactions, even when the user interacts in multiple coinjoins. What that means is that, multiple purchases at such stores can render mixing services useless as the off-chain information contained in the ad-trackers may
uniquely link transactions together (even without them leaking personally identifiable information) [81].

While mitigation of this attack is possible by blocking web-tracking through ad-blockers, the average user has limited possibilities due to the imperfection of the available tools.

2) CROSS-CHAIN EXCHANGE THROUGH SERVICES

With Bitcoin providing limited privacy guarantees, users wishing to remain anonymous are inclined to exchange their coins with more privacy-preserving ones. As argued in [82], however, to exchange coins from one cryptocurrency to another is not necessarily as "safe" as one might think.

For example, malicious users that attempt to exchange cryptocurrencies to Fiat-currencies will likely get caught, as most perform KYC for all users. Of course, one could try to utilise a non-KYC service, such as Shapeshift. As the paper shows, however, up to 90.54% of exchanges may leak data, which enables linking the user’s transaction from one ledger to another [82]. To mitigate this leakage, the authors in [82] note that the shielded pool of Zcash can be used directly into the Shapeshifter API; however, the feature appears to be discontinued as of writing. Also, the implementation of KYC service means that the user must prove her identity to the service, restricting her privacy.

V. DISCUSSION

We have presented several attacks on privacy in cryptocurrencies (summarized in Table 6). While these attacks highlight the vulnerabilities in cryptocurrencies, they also show how use of existing anonymity techniques does not necessarily guarantee anonymity either. This could be partly due to the linked nature of cryptocurrency transactions, where the coins have to exist before one can use them.

To improve privacy in cryptocurrencies, particularly in Bitcoin, several protocols have been proposed (summarized in Table 7). These proposals vary from new cryptographic schemes using zero-knowledge proofs and shielded addresses to network propagation mechanisms aiming to hide the IP-address of the source. However, as seen from Table 7, a lot of effort has been put into the area of cryptographic protocols. Still, as seen from Table 6, transaction propagation throughout the network poses a threat to the systems as a whole. While previous works have argued that Dining Cryptographers network or Mixnets can enforce anonymous broadcast between untrusting peers [37], [39], [45], the anonymous broadcast will, in many cases, still allow us to create the transaction graph and thereby perform several powerful attacks on the users’ anonymity. The actual value of anonymous broadcast may be apparent within the set of Zero-Knowledge transfer protocols, as those have no transaction graph.

Based on the literature review presented in this work, we conclude that the most popular protocols explored by the research community are Bitcoin, Monero, Zcash and MimbleWimble. We left out Zerocoin because of two reasons; i) the two-currency zero-knowledge structure is similar to Zerocash (which was an extension to Zerocoin), and ii) there

| TABLE 6. A holistic overview of the attack vectors limiting privacy in cryptocurrencies. |
|---|---|---|---|---|
| Objective | Method | Requirement | Papers |
| Clustering of addresses (outputs) | source-destination with known addresses | Transaction Graph | [27] |
| Re-identification of types | Pattern recognition | Transaction Graph & Known addresses | [29] |
| Link clusters to users | Scrape addressers from website | Transaction Graph & Known addresses | [20], [30] |
| Locating Source in P2P | Attack to bypass proxy-settings or access a specific server | A Tor server | [73] |
| Locating Source in Tor | Timing and relaying of messages | Nodes within the network | [71] |
| Double-spending and Locating Source of Transaction in Bitcoin | Timing and relaying of messages | Nodes within the network | [71] |
| Locating source of transaction | Utilise ISP information | Powerful attacker with ability to control the connection of the user | [74] |
| "Fishing" transactions and identities | Pulling "fishing" transactions that is related to a client | Verification | [75] |
| "Fishing" transactions and identities | Pulling "fishing" transactions that is related to a third-party tracker | Access to third-party tracker data | [81] |
| "Fishing" transactions and identities | Pulling "fishing" transactions and timing of transactions | No special requirements | [82] |

| TABLE 7. A summary overview of the proposals striving to mitigate the privacy-limitations from transaction graph- and network analysis, i.e. combining Table 2 and Table 5. |
|---|---|---|---|---|
| Method | Objective | Disadvantages | Papers |
| Centralised Mixing | Improve taint resistance | Require trust in a server. Limited anonymity if low server usage | [26], [27] |
| Decentralised Mixing | Improve taint resistance | Coordination with other participants, vulnerable to DoS and sybil attacks. | [36], [37], [40], [45] |
| Non-interactive Mixing | Improve taint resistance | Adversary set is limited in size, small blocks provide bad anonymity. | [46], [61] |
| Coin Swapping | Improve taint resistance | Require trust in the "seller", as he can leak the true transaction. | [39] |
| Ring signatures | Improve taint resistance | Limited anonymity-set. Rings can be "pumped". | [47], [49] |
| Stealth addresses | Hide receiver | Participant need to check if they are receiving incoming transactions. | [51], [52] |
| Confidential transactions | Hide amounts | Requires proofs for verification. Computationally expensive to verify in comparison to public (or SNAKES). | [47], [62] |
| ZK-proofs | Unlinkability | Requires a trusted setup, and stronger assumptions. | [11] |
| ZK-SNAKes | Unlinkability | Requires a trusted setup, and stronger assumptions. | [11], [12], [53], [54] |
| Mixnets | Hide IP | Vulnerable to DoS and Sybil attacks. | [79], [57], [58] |
| Tor & 12P | Hide IP | Vulnerable to DoS and Sybil attacks. | [79], [45] |

VOLUME 8, 2020
TABLE 8. Privacy features of the explored cryptocurrencies. ▼ = Easy to do. ▲ = difficult and ▼▼▼▼ = infeasible. A dotted circle, ◎, denotes the need for an active attacker who either logs the network or interacts with the attacked user.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Anonymity Set Size</th>
<th>Address Clustering</th>
<th>Sources &amp; Destinations</th>
<th>Transaction to IP</th>
<th>Transaction to Identity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bitcoin</td>
<td>▼▼▼▼</td>
<td>▼▼▼▼</td>
<td>▼▼▼▼</td>
<td>▼▼▼▼</td>
<td>▼▼▼▼</td>
</tr>
<tr>
<td>Monero</td>
<td>▼▼▼▼</td>
<td>▼▼▼▼</td>
<td>▼▼▼▼</td>
<td>▼▼▼▼</td>
<td>▼▼▼▼</td>
</tr>
<tr>
<td>Zcash</td>
<td>▼▼▼▼</td>
<td>▼▼▼▼</td>
<td>▼▼▼▼</td>
<td>▼▼▼▼</td>
<td>▼▼▼▼</td>
</tr>
<tr>
<td>XinDoo</td>
<td>▼▼▼▼</td>
<td>▼▼▼▼</td>
<td>▼▼▼▼</td>
<td>▼▼▼▼</td>
<td>▼▼▼▼</td>
</tr>
<tr>
<td>Wimble</td>
<td>▼▼▼▼</td>
<td>▼▼▼▼</td>
<td>▼▼▼▼</td>
<td>▼▼▼▼</td>
<td>▼▼▼▼</td>
</tr>
</tbody>
</table>

has been limited research interest in the currency, and only a few properties of their network have been analysed. Nevertheless, we find the current initiatives of Zerocoin interesting and expect more research to be conducted, especially with the examination of the Sigma Protocol [83] and the launch of Lelantus [9].

An overview of the four protocols can be found in Table 8. The table presents scoring in terms of easiness of performing an action, such as address clustering and linkage between sources and destinations, transactions to IPs and transactions to identities. The scores have been given holistically by the authors based on the analysis of the literature. We should note at this point that, while an attacker can link a transaction with an IP address (and identity) in the Zcash system, this does not enable them to see which coins were spent, but merely who published the transaction to the network.

Agreeing with Ian Miers,\(^\text{12}\) we note that decoy-based anonymity does leak information, and cannot truly hide the relationship between sources and destinations, like a Zero-Knowledge system. While ZK-SNArKs provide possibilities for anonymity in the cryptocoinsystem, from our analysis we conclude that the use of a trusted setup in an untrusted environment is problematic due to the potential backdoor that follows. Moreover, the highly complex cryptography that is required for their implementation makes bug-hunting and auditing of implementations very difficult and limited to a very small number of knowledgeable experts. This became apparent as Ariel Gabizon\(^\text{13}\) noted that a flaw in the Zcash system would allow an attacker to secretly mint currency at will. The flaw existed in the code for multiple months, but was unnoticed even by the Zcash team, due to the sophistication of the bug.\(^\text{14}\)

A. POSSIBLE RESEARCH DIRECTIONS

1) TRANSACTIONS PROPAGATION MECHANISMS

As the propagation of transactions poses a threat to both decoy-based and zero-knowledge proof systems, further research in this area is required, which should be tailored to the type of information retrievable from the various systems.

In particular, the development of non-interactive anonymous broadcasting can improve the network privacy for the zero-knowledge systems. However, anonymous broadcasting does not apply directly to decoy-based systems as those would still allow an observer to construct the transaction graph. For these systems to be sufficiently protected, the area of non-interactive transaction-aggregation throughout the propagation mechanism has to be explored. While promising proposals in this direction do exist,\(^\text{15}\) proof of substantial improvement is lacking.

2) SUCCINCT ZK PROOF SYSTEMS WITHOUT A TRUSTED SETUP\(^\text{16}\)

Within the latest few years, especially late 2019, several works have explored succinct zero-knowledge proofs without a trusted setup (also denoted as transparent setup). The most well-known of these proof systems are the STARK [84], Fractal [85] and Supersonic [86]. While these proof systems are promising, the size of their proofs are magnitudes larger than the current trusted setup, making them impractical for a blockchain setting (varying from tens to hundreds of KB for each proof, the current trusted setup SNArKs is 127 bytes [87], [88]). A group of proof systems exists between the transparent and the trusted setup, namely the SNORKs (PLONK [89], Sonic [90] and Marlin [91]). While SNORKs improve on the SNArK, in the sense that setups can be reused for multiple circuits, incentivising thus the one-off creation of a larger setup, they still require a trusted setup.

We believe that further research into closing the gap between trusted and transparent succinct zero-knowledge proof systems can significantly improve the privacy of deployed solutions, while mitigating the risks that follow the trusted setup.

3) SPECIALISED TRUSTLESS ZERO-KNOWLEDGE PROOFS

While the generalised proof systems provide promising directions for the blockchain space as a whole, the area of specialised proofs relevant for electronic cash (e.g., range- and 1-of-many proofs, already supports practical implementations without trusted setups (i.e., Sigma Protocol [83] and Lelantus [9]). These protocols, however, seldom offer as strong anonymity guarantees as their ZK-SNArK counterparts, due to performance limitations [9]. The exploration of protocols in this direction is an interesting field that could lead to improved privacy in value-transfer with more relaxed assumptions and accessible cryptography.

A summary of the discussion and the analysis findings, based on the research questions posed in this study, are presented in Table 9.


\(^{13}\)Who presented at the Theory and practice of blockchains conference at Aarhus University

\(^{14}\)https://fortune.com/2019/02/05/zcash-vulnerability-cryptocurrency/, accessed 03-12-2019

\(^{15}\)https://github.com/mimblewimble/grin/blob/master/doc/dandelion/dandelion.md, accessed 02-10-2019

\(^{16}\)Succinct meaning that the verification time is polylog(n), where n is the size of the circuit.
TABLE 9. Summary of the analysis findings based on the posed research questions.

<table>
<thead>
<tr>
<th>Research Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) What are the existing protocols and techniques for enforcing privacy in current cryptocurrencies?</td>
<td>Confidentiality of transaction amounts is enforced through commitments and range proofs. The anonymity of the transactions is provided via: (i) hiding within a small subset of transactions (decoy-based) and (ii) hiding within a large set of coins (zero-knowledge). Transaction propagation mechanisms address the anonymity at the network layer, be it simple diffusion or more advanced propagation mechanisms such as Dandelion. Zero-knowledge systems provide privacy guarantees, but are hard to comprehend by the average user. Deceit-based methods are easier to understand but have limited anonymity-sets, which can be further reduced through network analysis or knowledge of previous transactions. An attacker that observes a network with enough nodes can locate the source, and link the transaction with the IP of the peer. Furthermore, a user utilizing Tor is vulnerable to relative reality attacks. To improve the network properties of current techniques, we need to investigate the area of transaction propagation mechanisms. Addressing the issues of transaction graph analysis requires specialized trustless zero-knowledge proofs or improvements to ZK-SNARKs without a trusted setup.</td>
</tr>
<tr>
<td>2) How well do these protocols and techniques enforce privacy, and how can they be attacked?</td>
<td></td>
</tr>
<tr>
<td>3) How can we improve privacy in cryptocurrencies?</td>
<td></td>
</tr>
</tbody>
</table>

VI. CONCLUSIVE REMARKS

In this work, we have performed a systematic literature review on privacy within the space of electronic currencies. We find that no deployed solution, within the reviewed space, provides strong anonymity-guarantees for an average user. Further, we find that the existing proposals allow passive or active attacks that significantly influence the anonymity of the system. Moreover, only few countermeasures are deployed to mitigate network analysis, and many systems are significantly vulnerable against an attacker that observes the network.

From our literature review, we conclude that the techniques used within Zero-Knowledge systems provide stronger anonymity than their decoy-based counterparts. Based on this, we believe that they could play significant part in the battle towards true anonymity.

Moving forward, we believe that the space of privacy-enhancing network propagation mechanisms is the lowest hanging fruit. Furthermore, we firmly believe that research into improved performance of transparent SNARK-like proof systems can support scalable currencies with strong privacy guarantees. We, however, note that this type of system is highly complicated, and believe more specialised and simple Zero-Knowledge protocols can provide practical and anonymous payments in the near future.

REFERENCES
