Abstract—Blockchain and cryptocurrencies have been widely deployed and used in our daily life. Although there are numerous works in the literature surveying technical challenges and security issues in blockchains, very few works focused on the anonymity guarantees provided in cryptocurrencies. In this work, we conduct a systematic survey on anonymity in cryptocurrencies with a clear categorization for the different tiers of anonymity offered in the various cryptocurrencies as well as their known weaknesses and vulnerabilities. We also study the techniques that have been used to achieve each tier of anonymity. Finally, we assess the current techniques, and present a forecast for the technological trends in this field.

Index Terms—Blockchains, Cryptocurrencies, Privacy, Anonymity

I. INTRODUCTION

For many blockchain applications, particularly cryptocurrencies, anonymity is a critical property expected from the underlying platform. It does not only obscure users’ identity, but it also ensures fairness, without which users may opt-out of participation due to plausible fear of unfair treatment. Inadequate anonymity guarantees can also result in malicious parties focusing their efforts on de-anonymized high-value targets, leaking business information, and undermining the negotiation position. Besides, inefficient anonymity protection can lead to targeted denial of service which can decrease the fungibility of the affected cryptocurrency and further cripple its efficacy as a currency.

Moreover, different privacy enhancement technologies may offer the same (or similar) level of user anonymity, but their characteristics, like footprint, and computational effort and time, have direct repercussions on the adopt-ability of the cryptocurrency, its scalability, and even its transaction fees. For example, when Monero adopted the use of Bulletproofs instead of Borromean range proofs, the size of the transactions was reduced, consequently minimizing the transaction fees.

In the past decade, Bitcoin, altcoins, and many other decentralized blockchain-based applications have been hot research topics. As such, there are numerous surveys on blockchain and cryptocurrencies in the literature, e.g., the work of Tschorsch [64], including several surveys on general security issues in cryptocurrencies like the Conti et al.’s survey [25]. However, most of these works merely list all the cryptocurrencies and compare their privacy features at an ad-hoc level. In this work, for the first time, we provide a systematic study on the blockchain anonymity by categorizing the level of anonymity into four different tiers. We also examine the known techniques that can be used to achieve the various anonymity guarantees. Besides, we discuss the vulnerabilities and weaknesses of the different anonymity techniques, compare their effectiveness, and forecast their technological trends in this field.

Contributions. Previous surveys consider general security issues and challenges in Bitcoin and cryptocurrencies with no particular focus on anonymity and privacy [25][39][26][34]. The only previous survey that focuses on privacy and anonymity is the work of Khalilov et al. [38]; however, it does not provide any classification of the different levels of anonymity in cryptocurrencies, nor does it explain the related anonymity techniques. On the contrary, in this work, we provide a systematic study on blockchain anonymity. More specifically, our contributions are summarised as follows.

(1) Present a novel categorization for the tiers of anonymity offered in the diverse cryptocurrencies.
(2) Examine the techniques used to achieve the different tiers of anonymity, and discuss their vulnerabilities and weaknesses.
(3) Compare the anonymity techniques and forecast their technological trends.

Roadmap. In Sec. II we discuss the four tiers of anonymity offered in cryptocurrencies. In Sec. III we list and describe the techniques that are used to achieve the aforementioned tiers and the known attacks and weaknesses concerning each technique. In Sec. IV we discuss and assess the anonymity in the various cryptocurrencies, and forecast related technological trends. Finally, Sec. V presents a summary of the related work.

II. PRIVACY TIERS

The offered anonymity in any cryptocurrency and blockchain application can be assessed by considering two characteristics: (1) the ability of the used anonymity scheme to break any possible linkage between transactions, and (2) its ability in hiding users’ identities (senders and receivers). Given these two characteristics, we define four different tiers of anonymity in cryptocurrencies as follows: (1) pseudonymity, (2) set anonymity, (3) full anonymity, and (4) confidential transactions. Below we describe each of these tiers.
Pseudonymity. This is the most primitive level of anonymity in cryptocurrencies. This level of anonymity is guaranteed in Bitcoin, and, as shall be seen in Sec. III-A, this level of anonymity is achieved by using pseudo-anonymous addresses.

Set anonymity. In set anonymity, the identity of the user is either 1 out of \( n \) possible identities. Set anonymity is achieved by using ring signatures [55] where \( n \) is equal to the size of the ring. Similarly, mixers provide set anonymity where \( n \) is equal to the number of inputs in the mixer.

Full anonymity. This level is provided when the sender can be any node, and the sent note or coin can be any unspent note. As shall be discussed, this level is attained in by using commitments and zero-knowledge proofs as in Zerocoin [48] and Zcash [33].

Confidential transactions. This level guarantees that the transacted amounts are hidden. We emphasize that we do not limit this tier to Monero’s confidential transactions [45], and include but any approach that hides or obfuscates the transferred amounts to thwarts transaction flow analysis. Since a cryptocurrency can, for example, guarantee set anonymity while offering confidential transactions at the same time, as is the case in Monero [5], this tier of anonymity can also be thought of as an anonymity feature rather than a separate level of privacy.

Fig. 1 shows a pie chart representing the distribution of 20 cryptocurrencies and protocols according to their tier of anonymity. Details of these currencies shall be shown later in Table IV.

III. PRIVACY TECHNIQUES

To achieve the four anonymity levels mentioned in Sec. II, cryptocurrencies implement various anonymity techniques. In this section, we discuss six major techniques: (1) pseudonymous addressing, (2) ring signatures, (3) mixers, (4) commitments, (5) zero-knowledge proofs, and (6) stealth addressing. Besides, we give example implementation in cryptocurrencies, and list the known attacks and weakness concerning each of the aforementioned techniques.

A. Pseudonymous Addressing

Pseudonymous addressing aims to preserve privacy by breaking the link between addresses and their owners’ real identities. As discussed in the following paragraphs, it is widely known that Bitcoin is an example implementation of pseudonymous addressing in cryptocurrencies.

Pseudonymity in Bitcoin. Bitcoin is the first cryptocurrency, and remains to be the most successful one with a market value of over $211 billion\(^1\) and more than 9000 participating nodes\(^2\). As explained in its whitepaper [50], Bitcoin preserves users’ anonymity through the use of pseudonymous addresses. Namely, a user’s address is the Base-58 encoding of the following 25-byte binary string:

\[
\epsilon = \text{RIPEMD}-160(\text{SHA-256}(\text{publicKey}))
\]

\[
\text{address} = \text{version}|\epsilon|\text{checksum}
\]

Where publicKey is the user’s public key, version is a one-byte value indicating the version, and checksum is the least significant four bytes of the following value [64], [69]:

\[
\text{checksum} = \text{SHA-256}(\text{SHA-256}(\epsilon))
\]

Further, although they do not entirely enhance anonymity [12], Bitcoin users are advised to take two protective measures. 1) Users can generate a new key pair for each transaction [50]. In fact a person generating a transaction will also generate a new key pair so that the change address is not linked to the originating address and is indistinguishable from the destination’s address. 2) In every transaction, the sender fully empties one or more accounts, (inputs), and creates one or more accounts, (outputs). This approach helps break the linkage between the user’s accounts [23]. Nonetheless, as discussed in the following subsection, addresses can still be clustered and using the aid of public services and websites, the real identity of the users can eventually be revealed as discussed in many publications [47], [54], [56], [64].

Attacks on Pseudonymous Addressing. Pseudonymous addressing provides a weak anonymity guarantee. In fact, it is mentioned on Bitcoin’s official website that Bitcoin is not anonymous [3]. Consequently, it is not surprising that various de-anonymization attacks have been proposed in the relevant literature. Overall, these attacks can be classified into two broad categories: 1) clustering and Bitcoin blockchain analysis and 2) exploitation of the Bitcoin P2P network and diffusion protocol. A summary of these attacks is shown in Table IV.

\(^1\)https://coinmarketcap.com/ on 08/08/2019.


\(^3\)Linking Bitcoin pseudonyms to the user’s IP does not only cause a privacy breach but also may allow attackers to launch DoS against that user’s IP. This is more relevant if this user is a vendor or a service provider.
Clustering and Bitcoin blockchain analysis.

As shown in Fig 2, this attack analyzes Bitcoin traffic flow to cluster the user’s addresses and transactions and link them to a public service, e.g. an exchange website. These services can then reveal the real identity behind the pseudonymous address. In general, there are two approaches to clustering: analysis of transactions’ inputs and outputs, and behavioural analysis.

Analyzing transactions’ inputs and outputs. Use two particular heuristics to cluster transactions: (i) Inputs in one transaction are likely owned by the same user. Different users can theoretically contribute inputs in the same transaction, but rarely do so. (ii) Newer output address can be assumed to be a change address that belongs to the user generating the transaction. These heuristics result in representations of Bitcoin addresses, transactions, and users/entities. The last step in traffic analysis is to map users from these representations to real-world identities. This step aims to establish ownership and can be accomplished using the aid of public services, e.g. online stores and exchange website, which usually have user-identifying information such as email addresses and even bank accounts. To demonstrate the effectiveness of the above heuristics, the authors of [27] attempted to track known Bitcoin thefts to exchange services. In summary, they were able to track 6 out of 7 thefts from the point of theft to an exchange service. Furthermore, if presented with legal subpoena, these services can reveal the real identities behind the exchange operation. More importantly, their success in tracking these thefts prove that even privacy-conscious users, who seek to further hide their identities by sending (or peeling) some of their funds to newly generated addresses, are prone to de-anonymization using these heuristics.

Behavioural Analysis. In this type of attacks, addresses and transactions are clustered based on behavioural attributes like their time, location, and amount. Androulaki et al. [12] used behavioural analysis to augment their clusters; namely, they considered the time of the transactions, the indices of the inputs in a transaction, and the transferred amount. Using these techniques, they succeeded to unveil about 40% of the users in their simulated Bitcoin network. Similarly, Ron et al. [57] used behavioural analysis to link the Bitcoin addresses that are believed to be related to the Silk Road marketplace[6]. Also, Dupont et al. [28] analyzed user’s spending habits to reveal Bitcoin users physical location by analyzing. In addition, they assessed their method by collecting 518 known charities’ Bitcoin addresses and physical locations, and comparing this data against their informed guesses, where their initial results show an accuracy of up to 72%.

Exploiting Bitcoin P2P network.

This family of attacks exploits the nature of Bitcoin P2P network to link pseudonymous addresses to IP addresses. The work of Koshy et al. [56] constitutes the first proposal to de-anonymize Bitcoin users by studying the relay pattern of transactions, and they were able to map between 252 and 1162 Bitcoin addresses to the IP addresses that likely own them. Similarly, the work in [55] attempts to develop a probabilistic model to identify transactions’ originators’ IP based on monitoring the nodes that first relay a given transaction. Moreover, the authors of [16] attempted to de-anonymize Bitcoin client, even those sitting behind NATs, by the set of entry nodes they connect to. According to their methodology, the attacker tries to connect to the majority of servers, and they argue that when the attacker receives the transaction from 2 to 3 entry nodes, he can map the transaction to a specific client with a very high probability.

To strengthen their anonymity, Bitcoin users may choose to use anonymization tools, such as Tor [7]; however, as shown by Biryukov et al. in [17], combining Bitcoin and Tor introduces a new attack vector. The authors in [17] explore the exploitation of Bitcoin P2P with Tor beyond the mere banning of Bitcoin clients from using Tor exit nodes as previously done in [16]. The crux of their attack depends on: 1) forcing Bitcoin clients to connect to the attacker’s Tor Exit nodes or directly to the attacker’s Bitcoin peers, and 2) fingerprinting clients by writing unique (possibly fake) exit nodes to the attacker’s Tor exit nodes.

In 2015, the Bitcoin community has responded to the aforementioned attacks by changing its transactions broadcasting protocol from a gossip-like trickle spreading protocol to a diffusion spreading protocol [70]. To assess the impact on anonymity, the authors of [29] studied the properties of the two broadcasting protocols and their effect on user anonymity, and concluded the two Bitcoin flooding protocols do not protect the user anonymity. Also, Mastan et al. attempted to de-anonymize Bitcoin users sitting behind Tor by studying the pattern of their sessions and constructing a session graph, which they were able to perform with a precision of 0.9 [43].

Other attacks

Goldfeder et al. [32] studied the effect of web trackers on Bitcoin users when shopping online, and concluded that trackers can uniquely identify transactions, link them to the user’s cookie and reveal the user’s real identity. Other tools and frameworks have been proposed for visual traffic analysis of the Bitcoin blockchain [61][13].

| TABLE I: Attacks on Bitcoin Pseudonymity. |

<table>
<thead>
<tr>
<th>Attack</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clustering</td>
<td>Analyzes Bitcoin traffic flow to cluster addresses and transactions and link them to public services, revealing real identities.</td>
</tr>
<tr>
<td>Behavioural Analysis</td>
<td>Uses user-identifying information like time, location, and amount to cluster addresses and transactions.</td>
</tr>
<tr>
<td>Exploiting P2P Network</td>
<td>Attacks exploit the nature of the P2P network to link pseudonymous addresses to IP addresses.</td>
</tr>
<tr>
<td>Other Attacks</td>
<td>Includes tracking user spending habits, using anonymization tools like Tor, and fingerprinting clients.</td>
</tr>
</tbody>
</table>

Ring Signatures

Ring signatures were introduced by Rivest et al. [55] extending the idea of group signatures that was proposed by Chaum and van Heyst [24]. In a group signature, there is a trusted group manager who can de-anonymize the other signers. On the contrary, ring signatures are ad-hoc with no trusted manager, and any signer can sign on behalf of the group. Specifically, a ring signature scheme consists of a tuple of algorithms $\mathcal{S} = (\text{Setup}, \text{KeyGen}, \text{Sign}, \text{Verify})$ as follows:

- $\text{param} \leftarrow \text{Setup}(1^\lambda)$ is the setup algorithm that takes as input the security parameter $1^\lambda$, and it outputs a system parameter param. The rest of the algorithms implicitly take param as an input.
- $(PK, SK) \leftarrow \text{KeyGen}(\text{param})$ is the key generation algorithm that takes as input the setup parameter param, and outputs a pair of public and secret keys $(PK, SK)$.
- $\sigma \leftarrow \text{Sign}(\mathcal{P}, SK, \ell, m)$ is the signing algorithm that takes as input a set of public keys $\mathcal{P} := \{PK_1, \ldots, PK_n\}$, the secret key SK, the index $\ell$ such that SK is the corresponding secret key of $PK_\ell$, and the message $m$, and it outputs the signature $\sigma$.
- $b \leftarrow \text{Verify}(\mathcal{P}, m, \sigma)$ is the verification algorithm that takes as input a set of public keys $\mathcal{P}$, the message $m$ and the signature $\sigma$, and it outputs $b = 1$ if only if the signature is valid.

Ring signatures have evolved since their proposal in three directions: (1) threshold ring signatures [21], (2) linkable ring signatures [40] [13] [41], and (3) traceable ring signatures [31] [30]. To further clarify the usage of ring signatures in blockchains, we explain below the signature scheme used in the CryptoNote protocol and the use of ring signatures in Monero’s RingCT.

The use of ring signatures in CryptoNote. As shown in...
Borromean ring signatures which were used to construct the first 1-out-of-n ring signature invented by Maxwell and Polestra [46] that is an optimization of the AOS ring signature by Abe et al. [11]. Borromean ring signature is used as the key image. As seen in Fig. 3, the verifier does not know any information beyond the fact that 1 out of the possibly one-time private key as

$$P \leftarrow \{P_j\}_{j=1}^k$$

Let $$P := \{P_j\}_{j=1}^k$$ be a set of public keys, and the signing algorithm also takes input as the secret key $$t \in \{k\}$$ such that $$P = g^t$$, $$\in \{k\}$$

Denote $$I := \text{hash}_g(P_t)$$ as the key image. In Fig. 3, the verifier need not know any information beyond the fact that 1 out of the possible $$k$$ signers generated the signature $$\sigma$$.

**Borromean ring signature in Monero’s RingCT**. Borromean ring signature is a 1-out-of-n signature invented by Maxwell and Polestra [46] that is an optimization of the AOS ring signature by Abe et al. [11]. Borromean ring signature is used in Monero’s RingCT to generate range proofs by generating a ring signature for each digit of the committed amount. This effectively hides the committed amount $$a$$ while proving its range $$a \in \{0, 2^{31} - 1\}$$. If an amount $$a$$ is encoded in 16 base-4 digits $$d_0, d_1, d_2, \ldots, d_{15}$$, the sender chooses 16 blinding factors $$x_i$$ and generates 16 commitments, one for each digit as follows:

$$C_i = x_iG + a_iH$$

where $$a_i = (4^{15-i} \times d_i)$$, $$i \in \{0, 15\}$$ and $$d_i \in \{0, 3\}$$.

After that, the sender generates 4 public keys $$C_{i,d}$$ for each digit $$d_i$$.
corresponding to the 4 possible values \( d \in \{0, 3\} \) as follows:

\[
C_{i,d} = C_1 - (4^{15-i} \cdot d)H
\]

This will generate 4 public keys \( C_{i,d} \) for each ring signature for which the signer/sender knows one private key, \( x_i \) corresponding to the public key that was generated for the actual committed value of the digit. For example, if the \( 3^{rd} \) most significant digit of the base-4-encoded \( a \) has a value of 1, that is \( d_2 = 1 \), then the signer would know the private key \( x_2 \) corresponding to the second public key in the \( 3^{rd} \) ring: \( C_{2,1} \) because:

\[
C_{2,1} = C_2 - (4^{15-2} \cdot 1)H
\]

\[
C_{2,1} = (x_2G + (4^{15-2} \cdot 1)H) - (4^{15-2} \cdot 1)H = x_2G
\]

By choosing the blinding factors \( x_0, x_1, \ldots, x_{15} \) so that they add up to \( x \) which is the blinding factor used for the overall commitment \( C \), any party can publicly verify that \( C = C_0 + C_{1} + \cdots + C_{15} \). However, no one can know which of the possible 4 values each commitment corresponds to, nor can they know which value in the range is committed to.

**Attacks on ring signatures.** Although ring signatures have evolved over time, there remains some weaknesses that can be exploited as listed in Table III.

C. Mixers

To address the privacy limitations in Bitcoin, there have been multiple proposals to break any linkage between senders and recipients by mixing users’ funds through coin-laundry services called mixers. These mixers are generally in the following three forms: 1) trusted centralized mixers were the 1st-generation of mixers and demand unreasonable trust of third-party services to mix the user’s coins. 2) smart-contract-like mixers in which multiple users agree to create a joint transaction to obfuscate inputs and outputs, e.g. CoinJoin \([45]\) and CoinShuffle \([58]\), and 3) decentralized mixers which are trust-free cryptographic extensions to Bitcoin, e.g. Zerocoin \([48]\) and CoinParty \([73, 74]\). It is important to note that this latter type of mixers does not represent fully-fledged anonymous zero-knowledge-proof currencies, which are discussed later in Sec. III-E. Instead, this type represents extensions on top of other currencies, and may not be practical for day-to-day usage. For example, Zerocoin is presented in \([48]\) as a decentralized mix that extends Bitcoin; however, its limited functionality and high computational cost do not allow it to be used for routine transactions. Table III shows an up-to-date list of mixers proposed in the literature.

**De-anonymity Attacks on Mixers.** Table III lists the lists of weaknesses of the different types of mixers.

### Table III: List of proposed mixers in literature.

<table>
<thead>
<tr>
<th>Type</th>
<th>Examples</th>
<th>Disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centralized Mixers</td>
<td>Cryptomix ([3])</td>
<td>Single point of failure</td>
</tr>
<tr>
<td></td>
<td>Bitcoin Fog ([4])</td>
<td>No deniability against the mix itself ([22])</td>
</tr>
<tr>
<td></td>
<td>BestMix ([7])</td>
<td>No proof of mixing</td>
</tr>
<tr>
<td></td>
<td>Mixcoin ([17])</td>
<td>Unreasonable trust of 3(^{rd}) party</td>
</tr>
<tr>
<td></td>
<td>Blindcoin ([22])</td>
<td>Possible trick</td>
</tr>
<tr>
<td></td>
<td>Obscuro ([64])</td>
<td>Uses trusted execution environment (TEE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Assumes no mis-implementation by mixer operator</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Anonymity set is limited by block size ([21])</td>
</tr>
<tr>
<td>Smart-contract-like Mixers</td>
<td>CoinJoin([43])</td>
<td>Vulnerable to Sybil attacks</td>
</tr>
<tr>
<td></td>
<td>CoinShuffle ([58])</td>
<td>Vulnerable to collusion between users in mix</td>
</tr>
<tr>
<td></td>
<td>Obscuro ([63])</td>
<td>No anonymity against smudgers users in mix</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Malicious users can disrupt mixing ([53])</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hot user determines the outcome of the shuffle ([51])</td>
</tr>
<tr>
<td>Decentralized Mixers</td>
<td>CoinParty ([23])</td>
<td>Longer mixing delay ([41])</td>
</tr>
<tr>
<td></td>
<td>Zerocoin ([48])</td>
<td>Assumption 2/3 of the peers are honest</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Anonymity level is related to number of minted coins (between a coin’s mint and its spend)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reveals the number of mint and spent coins ([48])</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reveals transferred denominations ([23])</td>
</tr>
</tbody>
</table>

D. Commitments

Commitments are widely used as references/pointers to some secret, which allows the secret owner to demonstrate the properties of such a secret using (non-interactive) zero-knowledge proofs. For instance, the user could commit the balance of his/her account, and then use zero-knowledge proofs to show the balance is within a certain range, e.g., larger than 0. In the blockchain context, two types of commitment schemes have been used: (i) additive homomorphic, e.g., Pedersen commitment and its variants and (ii) non-malleable, e.g., hash-based commitments.

**(Generalized) Pedersen commitment.** This commitment \([52]\) is used in many blockchain platforms, such as Monero. To make the commitment non-interactive, the commitment key is typically given as a common reference string. Let \((G, H) \in \mathbb{G}^2\) be the commitment key. To commit a message \(m \in \mathbb{Z}_q\), the committer picks a fresh randomness \(r \in \mathbb{Z}_q\) and outputs the commitment as \(c := rG + mH\). It is easy to see that the commitment is computationally binding and unconditionally hiding. In Monero, the group is instantiated from elliptic curve (the secp256k1 curve). In addition, Pedersen commitments are additively homomorphic, i.e., they preserve addition and commutativity, which enables the public verification that the sum of the hidden input values is equal to the hidden output values. For example, disregarding transaction fee for simplicity, if a transaction has three inputs \(a, b, d\), and two outputs \(e\) and \(g\) s.t. \(a + b + d = e + g\).

**Hash-based Commitments.** Hash-based commitments have been used in Zcash, due to its efficiency, which enables fast zk-SNARK at that time. Unlike Pedersen commitment, hash-based commitments are usually transparent in the sense that the setup process is public coin. Hence, unlike common reference-string-based schemes, the setup process is believed to be subversion resistant. To commit a message \(m \in \{0, 1\}^\lambda\), the committer picks a random coin \(r \in \{0, 1\}^\lambda\) and outputs the commitment as \(c := \text{hash}(m, r)\), where \(\lambda\) is the security parameter, say 256 in practice. In Zcash, the hash was instantiated from SHA-256; recently, they switched to group-based structure-preserving hash. More details can be found in the Zcash protocol specification \([33]\).

E. (Non-interactive) Zero Knowledge Proofs

Typically, the zero-knowledge proofs used in blockchains need to be publicly verifiable, which means they need to be
non-interactive. On the other hand, it is well-known that non-interactive zero-knowledge (NIZK) proofs cannot be realized in the standard model, a.k.a. plain model; therefore, all the non-interactive zero-knowledge proofs require some setup assumptions, such as common reference string, random oracle, etc. As mentioned before, in terms of subversion resistance, the random oracle model is more preferred by the community. In general, NIZK proofs have been used to achieve anonymity in cryptocurrencies in three ways:

1) Utilizing the existing scripts in current cryptocurrencies to extend these cryptos and break the linkage between the senders and the receivers. Zerocoin is an example of this methodology.

2) Devising new cryptographic structures to replace current inefficient structures. An example of this type is the use of bulletproofs to replace Borromean ring signatures in Monero RingCT’s rangeproofs.

3) Designing new ZKP-based cryptocurrencies that are fully anonymous like Zcash, which is an implementation of the Zerocash protocol.

In the following, we discuss two commonly used NIZK techniques.

zk-SNARK. Succinct non-interactive zero-knowledge argument of knowledge (zk-SNARK) has two very important properties: (i) succinctness and (ii) unbalanced. Succinctness means that the proof size is less than poly-logarithmic (or constant in this concrete case) w.r.t. the witness size. Unbalanced indicates that the verifier’s running time is much less than the statement execution time, i.e. poly-logarithmic (or constant in this concrete case). In the blockchain context, a zero-knowledge proof needs to be verified by a great number of verifiers; hence, unbalanced is a very desired property. However, the cost of the proof generation is usually very high, which limits its wide adoption.

zk-SNARK is used to achieve full anonymity in Zerocash – a digital currency that is decentralized, privacy-preserving, and efficient. To anonymize the sender, the receiver, and mask the transferred amount, Zerocash uses a zk-SNARK. Zcash is a cryptocurrency that implements the Zerocash protocol. The Zcash blockchain contains two sets: a set containing commitments cm, and a set containing nullifiers nf. Hence, the Zcash blockchain does not only contain a database of unspent transactions but a database of all transactions that ever existed. To each note, there is a cryptographically associated note commitment and a nullifier (so that there is a 1:1:1 relation between notes, note commitments, and nullifiers). Computing the nullifier requires the associated private spending key sk. It is infeasible to correlate the note commitment with the corresponding nullifier without knowledge of at least this spending key. An unspent valid note, at a given point on the blockchain, is one for which the note commitment has been publicly revealed on the blockchain prior to that point, but the nullifier has not.

The basis of the privacy properties of Zcash is that when a note is spent, the spender only proves that some commitment for it had been revealed, without revealing which one. This implies that a spent note cannot be linked to the transaction in which it was created. That is, from an adversary’s point of view, the set of possibilities for a given note input to a transaction includes all previous notes that the adversary does not control or know to have been spent.

Bulletproofs. Bulletproofs are shorter zero-knowledge proofs that were proposed in and are based on the work of Bootle et al. While SNARK requires the use of pairing-based cryptography, and bilinear pairing groups in particular, Bulletproofs are based on discrete log computation; hence, Bulletproofs are suitable for all elliptic-curve algorithms and can prove arbitrary arithmetic circuit. For the prover’s running time, Bulletproof is much faster than zk-SNARK; however, the verifier’s running time is typically similar to the prover’s running time, which is linear in the statement execution. It means that Bulletproofs cannot be used to achieve verifiable computation sourcing, as the verifier needs to spend an equal amount of time to verify the proof. In Monero, Bulletproofs substantially reduce the size of transactions by replacing Borromean signature in generating range proofs.

Disadvantages of zero-knowledge proofs. Zero-knowledge proofs (ZKP) can provide strong anonymity guarantees, as in Zcash and Zerocoin; however, they suffer a few disadvantages. First, while various work in literature studied their subversion resistance, it is proven that non-interactive ZKPs for general NP language must require some trusted setup assumptions, such as, the security parameter’s generation ceremony in Zcash, which are susceptible to subversion. Also, the proof generation and verification can be computationally inefficient. More importantly, the prover’s efficiency is far from being practical for large-scale statement and/or verifiable computation.

F. Stealth Addressing

Stealth addressing is a technique proposed as part of the CryptoNote protocol to hide the recipient’s identity (or address). In short, the sender, Alice, uses Diffie-Hellman exchange to compute a shared secret and generate a one-time destination address that can only be identified by the intended recipient, Bob. Specifically, let’s assume Bob’s public key is the pair $\{A, B\}$ that corresponds to his private key $(a, b)$, such that $A = aG$, and $B = bG$, where $G$ is the base point of the used elliptic curve. In this case, Alice generates a random number $r$, and a one-time address $P = H_s\{rA\}G + B$, where $H_s$ is a collision-resistant cryptographic hash function. Along with $P$, Alice sends $R = rG$ as part of the transaction. Bob checks every transaction using his private key $(a, b)$ and computing $P' = H_s\{aR\}G + B$. If the transaction is destined for Bob, then $P' = P$.

More generally, the stealth addressing technique can be extended to a post-quantum secure stealth addressing scheme by replacing the underlying primitives.
TABLE IV: Categorization of 20 cryptocurrencies and protocols according to their tier of privacy and used techniques. (1) Zero-knowledge proofs. (2) The anonymity of a spent coin is relative to the number of minted coins before spending this coin. (3) This is with respect to shielded transactions.

<table>
<thead>
<tr>
<th>#</th>
<th>currency</th>
<th>anonymity tier</th>
<th>Anonymity Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bitcoin</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>Ethereum</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>Ethereum Classic</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>4</td>
<td>Bitcoin Cash</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>5</td>
<td>Bitcoin Diamond</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>6</td>
<td>Litecoin</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>7</td>
<td>Cardano</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>8</td>
<td>IOTA</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>9</td>
<td>Dogecoin</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>10</td>
<td>NEM</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>11</td>
<td>Nano</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>12</td>
<td>Zcash</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>13</td>
<td>Zerocoin</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>14</td>
<td>Monero</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>15</td>
<td>Zcash</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>16</td>
<td>Dash</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>17</td>
<td>Bitcoin</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>18</td>
<td>Monero</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>19</td>
<td>Zerocoin</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>20</td>
<td>Zcash</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Fig. 4: Timeline of Monero privacy enhancements [9][5].

IV. DISCUSSION

Cryptocurrencies, tiers of anonymity, and techniques. Table IV summarizes the tiers of anonymity offered in 20 currencies and the techniques they implement to achieve privacy and anonymity.

Comparison between different schemes. As shown in Table IV, one can conclude that ZKP and commitments are used to achieve the highest tier of anonymity; full anonymity. However, these two techniques can increase the computational cost, and may require a trusted setup. Therefore, anonymity schemes shouldn’t be assessed only by considering the level of privacy they provide but by jointly examining three factors: 1) the level of anonymity provided by using the scheme, 2) the scheme’s computational efficiency, and 3) the extent of needed trust to use the technique.

Technology trend. The future technological trend in privacy mechanisms is best exemplified by Monero’s evolution since its inception. As shown in Fig. 4, Monero was initially based on the CryptoNote protocol which uses linkable ring signatures and then evolved over time and adapted the use of Bulletproofs to replace Borromean signatures in its RingCT’s rangeproofs. This demonstrates the continuous quest for cryptocurrencies to adapt privacy schemes that: 1) offer a higher tier of anonymity, 2) require less generation and verification time, 3) produce more succinct proofs, 4) do not require trusted setup, and 5) possibly, result in minimal transaction fees. The design of new anonymity schemes should also consider their impact on the forkability of the cryptocurrency [68].

Open problem. There are some known intrinsic vulnerabilities concerning anonymization in cryptocurrencies and blockchains in general. One of these vulnerabilities is leaking the user’s IP address and timestamp whenever a user broadcasts a transaction. This can be exploited, as demonstrated in many works [36][16][17][29], to de-anonymize the users regardless of the specific blockchain application they are using. Furthermore, even when using an anonymization tool, like Tor, users can still be de-anonymized as previously discussed in Sec. III.

V. RELATED WORK

In 2015, Bonneau et al. [18] presented the first SoK to generally survey issues related to Bitcoin and altcoin. With specific focus on Bitcoin, Conti et al. systematically surveyed security and privacy issues in Bitcoin, and listed respective countermeasures [25]. Other surveys on general security issues in blockchains include the work of Li et al. [39], the work of Dasgupta et al. [26], and the survey of Joshi at al. [34]. The work of Khalilov et al. [38] presents a comprehensive survey of all studies related to anonymity in Bitcoin, and studies related to anonymity in schemes that are improvements to Bitcoin and other schemes that are alternative to Bitcoin. However, unlike our work, the work of Khalilov et al. [38] does not categorize cryptocurrencies in terms of the level of anonymity they provide, nor does it thoroughly explain the used privacy techniques.

VI. CONCLUSION

In this work, we presented a high-level categorization of the anonymity guarantees offered in cryptocurrencies, and showed that most cryptocurrencies, 15 out of 20, still use the most primitive level of anonymity; pseudonymity. Besides, we explained the used anonymity schemes and their weaknesses. Finally, we assessed the different anonymity techniques and presented a forecast for their future technological trends.
values.txt (Last accessed 25-June-2019).


