SoK: A Systematic Study of Anonymity in Cryptocurrencies

Nasser Alsalami
Lancaster University - UK
n.alsalami@lancaster.ac.uk

Bingsheng Zhang
Lancaster University - UK
b.zhang2@lancaster.ac.uk

***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 asses 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 [45].

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 [22] instead of Borromean [46] range proofs, the size of the transactions was reduced, consequently minimizing the transaction fees [51].

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 asses 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.

978-1-7281-2319-6/19/$31.00 ©2019 IEEE

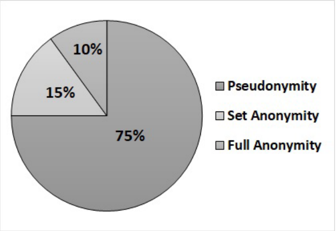

Fig. 1: Distribution of 20 cryptocurrencies and protocols according to their tier of anonymity.

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 currencies and implementations according to their tier of anonymity. Details of these currencies shall are 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.

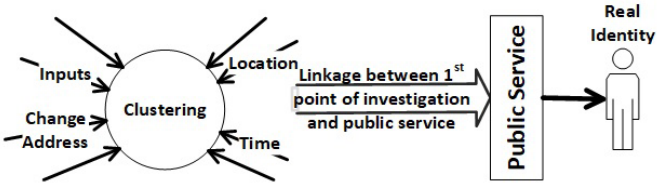

Fig. 2: Representation of clustering attacks on Bitcoin.

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 preservers users' anonymity through the use of pseudonymous addresses. Namely, a user's address is the Base-58 encoding of the the following 25-byte binary string:

$$\epsilon = \mathsf{RIPEMD\text{-}160}(\mathsf{SHA\text{-}256}(\mathsf{publicKey}))$$
$$\mathsf{address} = \mathsf{version}||\epsilon||\mathsf{checksum}$$

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]:

$$\mathsf{checksum} = \mathsf{SHA\text{-}256}(\mathsf{SHA\text{-}256}(\epsilon))$$

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 I.

[1] https://coinmarketcap.com/ on 08/08/2019.

[2] https://bitnodes.earn.com/ at 13:22:21 UTC 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.

1) 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 [50].* 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* [47][54][56][64]. 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 [54]. To demonstrate the effectiveness of the above heuristics, the authors of [47], 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%.

2) 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. [36] constitutes the first proposal to deanonymize Bitcoin users by studying the relay patterns of transactions, and they were able to map between 252 and 1162 Bitcoin addresses to the IP addresses that likely own them [3]. Similarly, the work in [35] 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 deanonymize 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.[4] 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) addresses to the target's address table. 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 asses 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 [42] .

3) 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][14].

TABLE I: Attacks on Bitcoin Pseudonymity.

B. 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 $\mathcal{S} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ as follows:

- $\mathsf{param} \leftarrow \mathsf{Setup}(1^\lambda)$ is the setup algorithm that takes as input the security parameter 1^λ, and it outputs a system parameter param. The rest of the algorithms implicitly take param as an input.
- $(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathsf{KeyGen}(\mathsf{param})$ is the key generation algorithm that takes as input the setup parameter param, and outputs a pair of public and secret keys $(\mathrm{PK}, \mathrm{SK})$.
- $\sigma \leftarrow \mathsf{Sign}(\mathcal{P}, \mathrm{SK}, \ell, m)$ is the signing algorithm that takes as input a set of public keys $\mathcal{P} := \{\mathrm{PK}_1, \ldots, \mathrm{PK}_n\}$, the secret key SK, the index ℓ such that SK is the corresponding secret key of PK_ℓ, and the message m, and it outputs the signature σ.
- $b \leftarrow \mathsf{Verify}(\mathcal{P}, m, \sigma)$ is the verification algorithm that takes as input a set of public keys $\mathcal{P}$, the message m and the signature σ, 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

[4] A list of current active Tor exit nodes can be found in: https://torstatus.blutmagie.de/

Vulnerabilities and Weaknesses of Ring Signatures	
Weakness	**Description**
Deducibility due to 0-mixin coins.	The authors of two independent works [49][37] described two weaknesses in Monero's ring signature. The first weakness is the deducibility of the real spent input as a result of referencing outputs that have been provably spent or consumed in previous 0-mixin transactions, where *mixin* refers to the number of decoy outputs that are referenced in a transaction to anonymize the real to-be-consumed output. Specifically, they found that 0-mixin transactions don't only de-anonymize the output they reference but have a cascading effect which can result in de-anonymizing other transactions with mixin ≥ 1.
Identifying real inputs using temporal analysis.	The second weakness described in [49], similarly mentioned in [37], is related to the sampling of mixins (or chaff coins). Namely, the authors of [49] have found that, about 80% of the time, the real consumed output is the newest created coin.
Identifying real inputs using inference.	Beyond the effect of zero-mixin transactions and temporal analysis, Yu et al. [71] devised a set of security games called "The Sun-Tzu Survival Problem" to model untraceability in CryptoNote-based cryptocurrencies, and explained that the sampling strategy of decoy mixins can enable attackers to infer the real to-be-spent inputs. Similarly, another work described *closed set* attack which is based on the observation that if the number of referenced outputs (or public keys) in a set of transaction inputs is equal to the number of inputs, then these referenced outputs must be the real consumed outputs in these inputs and mere decoy mixins whenever referenced outside of this set [72].
Flooding the network with attacker-generated outputs.	The authors of [66] described two attacks on Monero ring signature. The first attack is an extension to the discussion from [62] and based on flooding the network with outputs that are generated by the attacker(s) and addressed to their own addresses. Therefore, if these outputs are referenced as decoy outputs, i.e. mixins, in any ring signature, the attacker(s), who passively monitors the signatures, can rule out their outputs and hence decrease the anonymity of the signer. If, for example, a transaction references n outputs/coins, i.e. has a mixin of size n, and m of which are generated by the attacker, then the effective mixin size is reduced to $(n - m)$.
Subverted sampling of outputs.	The second attack described in [66] is an active version of the previous attack. Namely, the attacker mis-implements wallets to sample his outputs when generating ring signatures. Therefore, the attacker who continuously monitors all transactions, can de-anonymize the real spent outputs.
Anonymity reduction by observing identical UPID.	The authors of [67] describe an anonymity reduction attack on Monero transactions by observing identical UPID in different transactions. Namely, they state that if a transaction T_a has a UPID U_a and generates some output O_a, and a latter transaction T_b that uses the same UPID U_a and references O_a as part of its mixins, then O_a is likely to be the real spent output in T_b and not a mere decoy output.
De-anonymization by new forks	Wijaya et al. [68] demonstrated that Monero hard forks can lead to traceability of the real spent outputs when the user spends the coins in the original blockchain and the newly forked blockchain.
Time and size	Borromean ring signatures which were used to construct *rangeproofs* in Monero RingCT resulted in rangeproofs that are several kilobytes in size and take milliseconds to verify [53]. Hence, the crypto community has been looking for a more succinct and faster to verify which eventually resulted in devising *Bulletproofs* [22] as discussed in Sec. III-E.

TABLE II: List of attacks on ring signatures.

its whitepaper [59], CryptoNote's signature uses a slightly modified version of the traceable ring signature scheme proposed by Fujisaki et al. [31]. According to this protocol, the payer generates a one-time public key $R := g^r$ and computes the address $T := g^{\mathsf{hash}_p(A^r)} \cdot B$. In this case, the payee is able to compute the corresponding one-time private key as $t := \mathsf{hash}_p(R^a) + b$. Note that the one-time ring signature scheme is transformed from the OR-composition of Schnorr's identification Sigma protocols. Also, the protocol has a LNK algorithm to link any two signatures produced by the same signing key, which is important to prevent double spending. For clarity, the signing algorithm is depicted in Fig. 3. Let $\mathcal{P} := \{P_j\}_{j=1}^k$ be a set of public keys, and the signing algorithm also takes input as the secret key t_ℓ such that $P_\ell = g^{t_\ell}$, $\ell \in [k]$. Denote $I := \mathsf{hash}_g(P_\ell)$ as the key image. As seen in Fig. 3, the verifier does not know any information beyond the fact that 1 out of the possible k signers generated the signature σ.

```
function Sign($\{P_j\}_{j=1}^k, t_\ell, \ell, m$):
  • Set $I := \mathsf{hash}_g(P_\ell)$;
  • For $j \in [k]$, pick $q_j \overset{\$}{\leftarrow} \mathbb{Z}_p$;
  • For $j \in [k], j \neq \ell$, pick $w_j \overset{\$}{\leftarrow} \mathbb{Z}_p$;
  • For $j \in [k]$:
      – Set $L_j := g^{q_j}$ if $j = \ell$;
        Set $L_j := g^{q_j} \cdot P_j^{w_j}$ if $j \neq \ell$;
      – Set $R_j := (\mathsf{hash}_g(P_j))^{q_j}$ if $j = \ell$;
        Set $R_j := (\mathsf{hash}_g(P_j))^{q_j} \cdot I^{w_j}$ if $j \neq \ell$;
  • Set $c := \mathsf{hash}_p(m, L_1, \dots, L_k, R_1, \dots, R_k)$;
  • For $j \in [k]$:
      – Set $c_j := w_j$ if $j \neq \ell$;
        Set $c_j := c - \sum_{j=1}^k c_j$ if $j = \ell$;
      – Set $r_j := q_j$ if $j \neq \ell$;
        Set $r_j := q_\ell - c_\ell t_\ell$ if $j = \ell$;
  • Return $\sigma := (I, c_1, \dots, c_k, r_1, \dots, r_k)$.
end function
function Verify($\{P_j\}_{j=1}^k, m, \sigma$):
  • For $j \in [k]$:
      – Set $L'_j := g^{r_j} \cdot P_j^{c_j}$;
      – Set $R'_j := (\mathsf{hash}_g(P_j))^{r_j} \cdot I^{c_j}$;
  • Check if $\sum_{j=1}^k c_j \overset{?}{=} \mathsf{hash}_p(m, L'_1, \dots, L'_k, R'_1, \dots, R'_k)$
end function
```

Fig. 3: CryptoNote Ring Signature: signing and verification algorithms

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 *rangeproofs* 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 $d_0 d_1 d_2 \dots d_{15}$, the sender chooses 16 blinding factors x_i and generates 16 commitments, one for each digit as follows:

$$C_i = x_i G + a_i H$$

where $a_i = (4^{15-i} * d_i)$, $i \in [0, 15]$ and $d_i \in [0, 3]$. After that, the sender generates 4 public keys $C_{i,d}$ for each digit d_i

Type	Examples	Disadvantage
Centralized Mixers	CryptoMixer [4] Bitcoin Fog [2] BestMixer [1] Mixcoin [19] Blindcoin [65]	Single point of failure
		No deniability against the mix itself [73]
		No prove of mixing
		Unreasonable trust of 3^{rd} party
		Possible theft
	Obscuro [63]	Uses trusted execution environments (TEE) Assumes no mis-implementation by mixer operator Anonymity set is limited by block size [63]
Smart-contract-like Mixers	CoinJoin⋆ [43] CoinShuffle† [58] Coinswap [44]	⋆No anonymity against insiders (users in mix)
		Vulnerable to Sybil attacks
		Vulnerable to collusion between users in mix
		Anonymity set = the number of users in mix [73]
		†last user determines the outcome of the shuffle [73]
		Malicious users can disrupt mixing [63][1]
Decentralized Mixers	CoinParty [73]	Longer mixing delay [73]
		Assumption 2/3 of the peers are honest
	Zerocoin [48]	Anonymity level is related to number of minted [48] coins (between a coin's mint and its spend)
		Reveals the number of mint and spent coins [48]
		Reveals transferred denominations [48]

TABLE III: List of proposed mixers in literature. 1: Users can join mixing and then abort to disrupt the operation.

corresponding to the 4 possible values $d \in \{0, 3\}$ as follows:

$$C_{i,d} = C_i - (4^{15-i} * d)H$$

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:

$$C_{2,1} = C_2 - (4^{15-2} * 1)H$$

$$C_{2,1} = (x_2 G + (4^{15-2} * 1)H) - (4^{15-2} * 1)H = x_2 G$$

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 II.

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 1^{st}-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 [43] 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-anonymization Attacks on Mixers. Table III lists the known weaknesses of the different types of mixers.

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 $c := r \cdot G + m \cdot h$. 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, and d, and two outputs e and g s.t. $a + b + d = e + g$.

Hash-based Commitments. Hash-based commitments has 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 $m \in \{0, 1\}^*$, the committer picks a random coin $r \in \{0, 1\}^\lambda$ and outputs the commitment as $c := \mathsf{hash}(m, r)$, where λ 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 [8] which is an implementation of the Zerocash protocol [60].

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 [60] – 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 [8] 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* $\mathsf{a_{sk}}$. 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 [22] and are based on the work of Bootle et al. [20]. 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 [46] 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 [15][10], 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 [59] to hide the recipient's identity (or address). In short, the sender, Alice, uses Diffie-Hellman exchange [27] 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 $P = \mathcal{H}_s\{rA\}G + B$, where $\mathcal{H}_s$ 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 $P' = \mathcal{H}_s\{aR\}G + B$. If the transaction is destined for Bob, then $P' = P$.

More generally, the stealth addressing technique can generalised to any non-interactive key exchange (NIKE) together with the public key system. Hence, the above scheme can be easily extended to a post-quantum secure stealth addressing scheme by replacing the underlying primitives.

		Anonymity Tier				Anonymity Technique						
#	crypto-currency	pseudo-anonymity	set anonymity	full anonymity	+CT	pseudonymous addresses	ring signature	mixers	commitments	ZKP [1]	bullet-proofs	stealth addressing
1	Bitcoin	✓				✓						
2	Ethereum	✓				✓						
3	Ethereum Classic	✓				✓						
4	Bitcoin Cash	✓				✓						
5	Bitcoin Diamond	✓				✓						
6	Litecoin	✓				✓						
7	Cardano	✓				✓						
8	IOTA	✓				✓						
9	Dogecoin	✓				✓						
10	NEM	✓				✓						
11	Nano	✓				✓						
12	Lisk	✓				✓						
13	Waves	✓				✓						
14	Tether	✓				✓						
15	USD Coin	✓				✓						
16	Dash		✓				✓					
17	Bytecoin		✓				✓					✓
18	Monero		✓		✓		✓				✓	✓
19	Zerocoin			✓[2]				✓	✓	✓		
20	Zcash			✓[3]	✓				✓	✓		

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*.

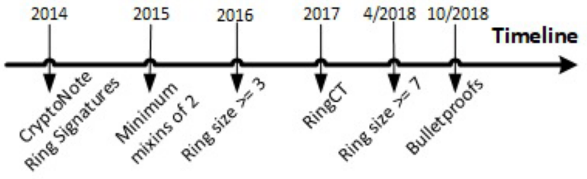

Fig. 4: Timeline of Monero privacy enhancements [9][5].

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 anonymity 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.

VII. References

[1] BestMixer. Available Online: https://bestmixer.io (Last accessed 22-Feb-2019).

[2] Bitcoin Fog. Available Online: https://bitcoinfog.info (Last accessed 12-Aug-2019).

[3] Bitcoin Project. Available Online: https://bitcoin.org/en/ (Last accessed 22-Aug-2019).

[4] CryptoMixer. Available Online: https://cryptomixer.io/ (Last accessed 15-Aug-2019).

[5] Monero Project. Available Online: https://github.com/monero-project/monero (Last accessed 16-Aug-2019).

[6] Silk Road (marketplace). Available Online: https://en.wikipedia.org/wiki/Silk_Road_(marketplace) (Last accessed 22-Aug-2019).

[7] Tor. Available Online: https://www.torproject.org/ (Last accessed 10-Aug-2019).

[8] Zcash. Available Online: https://z.cash (Last accessed 15-Aug-2019).

[9] A. Mackenzie, S. Noether and Monero Core Team. Improving Obfuscation in the CryptoNote Protocol, 1 2015. Available Online: https://lab.getmonero.org/pubs/MRL-0004.pdf (Last accessed 01-Jan.-2019).

[10] Behzad Abdolmaleki, Karim Baghery, Helger Lipmaa, and Michał Zając. A subversion-resistant snark. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017*, pages 3–33, Cham, 2017. Springer International Publishing.

[11] Masayuki Abe, Miyako Ohkubo, and Koutarou Suzuki. 1-out-of-n signatures from a variety of keys. In *ASIACRYPT 2002*, 2002.

[12] Elli Androulaki, Ghassan O. Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. Evaluating user privacy in bitcoin. In Ahmad-Reza Sadeghi, editor, *Financial Cryptography and Data Security*, pages 34–51, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[13] Man Ho Au, Sherman S. M. Chow, Willy Susilo, and Patrick P. Tsang. Short linkable ring signatures revisited. In Andrea S. Atzeni and Antonio Lioy, editors, *Public Key Infrastructure*, pages 101–115, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[14] G. D. Battista, V. D. Donato, M. Patrignani, M. Pizzonia, V. Roselli, and R. Tamassia. Bitconeview: visualization of flows in the bitcoin transaction graph. In *2015 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pages 1–8, Oct 2015.

[15] Mihir Bellare, Georg Fuchsbauer, and Alessandra Scafuro. Nizks with an untrusted crs: Security in the face of parameter subversion. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016*, pages 777–804, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[16] Alex Biryukov, Dmitry Khovratovich, and Ivan Pustogarov. Deanonymisation of clients in bitcoin p2p network. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 15–29, New York, NY, USA, 2014. ACM.

[17] Alex Biryukov and Ivan Pustogarov. Bitcoin over tor isn't a good idea. In *2015 IEEE Symposium on Security and Privacy*, pages 122–134, Washington, DC, USA, 2015. IEEE Computer Society.

[18] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In *2015 IEEE Symposium on Security and Privacy*, pages 104–121, May 2015.

[19] Joseph Bonneau, Arvind Narayanan, Andrew Miller, Jeremy Clark, Joshua A. Kroll, and Edward W. Felten. Mixcoin: Anonymity for bitcoin with accountable mixes. In Nicolas Christin and Reihaneh Safavi-Naini, editors, *Financial Cryptography and Data Security*, pages 486–504, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[20] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016*, pages 327–357, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[21] Emmanuel Bresson, Jacques Stern, and Michael Szydlo. Threshold ring signatures and applications to ad-hoc groups. In Moti Yung, editor, *Advances in Cryptology — CRYPTO 2002*, pages 465–480, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[22] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 315–334, May 2018.

[23] Vitalik Buterin. Privacy on the blockchain. Available Online : https://blog.ethereum.org/2016/01/15/privacy-on-the-blockchain/(Last accessed 09-Oct.-2018).

[24] David Cham and E. van Heyst. Group signatures. In D. W. Davies, editor, *Eurocrypt 1991*, pages 257–65. Springer-Verlag, 1991.

[25] M. Conti, E. Sandeep Kumar, C. Lal, and S. Ruj. A survey on security and privacy issues of bitcoin. *IEEE Communications Surveys Tutorials*, 20(4):3416–3452, 2018.

[26] Dipankar Dasgupta, John M. Shrein, and Kishor Datta Gupta. A survey of blockchain from security perspective. *Journal of Banking and Financial Technology*, 3(1):1–17, Apr 2019.

[27] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, November 1976.

[28] Jules DuPont and Anna Cinzia Squicciarini. Toward de-anonymizing bitcoin by mapping users location. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, CODASPY '15, pages 139–141, 2015.

[29] Giulia C. Fanti and Pramod Viswanath. Anonymity properties of the bitcoin P2P network. *CoRR*, abs/1703.08761, 2017.

[30] Eiichiro Fujisaki. Sub-linear size traceable ring signatures without random oracles. In Aggelos Kiayias, editor, *Topics in Cryptology – CT-RSA 2011*, pages 393–415, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[31] Eiichiro Fujisaki and Koutarou Suzuki. Traceable ring signature. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *Public Key Cryptography – PKC 2007*, pages 181–200, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[32] Steven Goldfeder, Harry Kalodner, Dillon Reisman, and Arvind Narayanan. When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies. *Proceedings on Privacy Enhancing Technologies*, 2018(4):179 – 199, 2018.

[33] Daira Hopwood, Sean Bowe, Taylor Hornby, and Nathan Wilcox. Zcash protocol specification, 2018. Available Online: https://cryptoverze-0m5mfism.netdna-ssl.com/wp-content/uploads/2019/01/z-cash-zec-whitepaper.pdf (Last accessed 22-Aug-2019).

[34] Archana Prashanth Joshi. A survey on security and privacy issues of blockchain technology. *Mathematical Foundations of Computing*, 2018.

[35] P. L. Juhász, J. Stéger, D. Kondor, and G. Vattay. A bayesian approach to identify bitcoin users. *PLOS ONE*, 13(12):1–21, 12 2018.

[36] Philip Koshy, Diana Koshy, and Patrick McDaniel. An analysis of anonymity in bitcoin using p2p network traffic. In Nicolas Christin and Reihaneh Safavi-Naini, editors, *Financial Cryptography and Data Security*, pages 469–485, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[37] Amrit Kumar, Clément Fischer, Shruti Tople, and Prateek Saxena. A traceability analysis of monero's blockchain. In Simon N. Foley, Dieter Gollmann, and Einar Snekkenes, editors, *Computer Security – ESORICS 2017*, pages 153–173, Cham, 2017. Springer International Publishing.

[38] M. C. Kus Khalilov and A. Levi. A survey on anonymity and privacy in bitcoin-like digital cash systems. *IEEE Communications Surveys Tutorials*, 20(3):2543–2585, 2018.

[39] Xiaoqi Li, Peng Jiang, Ting Chen, Xiapu Luo, and Qiaoyan Wen. A survey on the security of blockchain systems. *Future Generation Computer Systems*, 2017.

[40] Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups. In Huaxiong Wang, Josef Pieprzyk, and Vijay Varadharajan, editors, *Information Security and Privacy*, pages 325–335, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[41] Joseph K. Liu and Duncan S. Wong. Linkable ring signatures: Security models and new schemes. In Osvaldo Gervasi, Marina L. Gavrilova, Vipin Kumar, Antonio Laganà, Heow Pueh Lee, Youngsong Mun, David Taniar, and Chih Jeng Kenneth Tan, editors, *Computational Science and Its Applications – ICCSA 2005*, pages 614–623, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[42] Indra Deep Mastan and Souradyuti Paul. A new approach to deanonymization of unreachable bitcoin nodes. In Srdjan Capkun and Sherman S. M. Chow, editors, *Cryptology and Network Security*, pages 277–298, Cham, 2018. Springer International Publishing.

[43] G. Maxwell. Coinjoin: Bitcoin privacy for the real world, 8 2013. Available Online : https://bitcointalk.org/index.php?topic=279249.0 (Last accessed 25-Oct.-2018).

[44] Greg Maxwell. Coinswap: Transaction graph disjoint trustless trading. Available Online: https://bitcointalk.org/index.php?topic=321228.0 (Last accessed 12-Aug-2019).

[45] Greg Maxwell. Confidential transactions. Available Online: https:

//people.xiph.org/~greg/confidential_values.txt (Last accessed 25-June-2019).

[46] Gregory Maxwell and Andrew Poelstra. Borromean Ring Signatures, 2015. Available Online: http://diyhpl.us/~bryan/papers2/bitcoin/Borromean%20ring%20signatures.pdf (Last accessed 07-Feb-2018).

[47] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage. A fistful of bitcoins: Characterizing payments among men with no names. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 127–140, New York, NY, USA, 2013. ACM.

[48] I. Miers, C. Garman, M. Green, and A. D. Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In *2013 IEEE Symposium on Security and Privacy*, pages 397–411, 5 2013.

[49] Malte Möser, Kyle Soska, Ethan Heilman, Kevin Lee, Henry Heffan, Shashvat Srivastava, Kyle Hogan, Jason Hennessey, Andrew Miller, Arvind Narayanan, and Nicolas Christin. An empirical analysis of traceability in the monero blockchain. *Proceedings on Privacy Enhancing Technologies*, 2018(3), 2018.

[50] Satoshi Nakamoto. A Peer-to-Peer Electronic Cash System, 2008. Available Online: https://bitcoin.org/bitcoin.pdf (Last accessed 05-Nov-2018).

[51] R.R. O'Leary. Monero Fees Fall to Almost Zero After 'Bulletproofs' Upgrade, 2018. Available Online: https://www.coindesk.com/monero-fees-fall-to-almost-zero-after-bulletproofs-upgrade (Last accessed 13-Aug-2019).

[52] Torben Pryds Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *Advances in Cryptology — CRYPTO '91*, pages 129–140, Berlin, Heidelberg, 1992. Springer Berlin Heidelberg.

[53] Andrew Poelstra. Bulletproofs: Faster Rangeproofs and Much More, 2018. Available Online: https://blockstream.com/2018/02/21/en-bulletproofs-faster-rangeproofs-and-much-more/ (Last accessed 13-Aug-2019).

[54] F. Reid and M. Harrigan. An analysis of anonymity in the bitcoin system. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pages 1318–1326, 10 2011.

[55] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT 2001*, pages 552–565, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[56] Dorit Ron and Adi Shamir. Quantitative analysis of the full bitcoin transaction graph. In Ahmad-Reza Sadeghi, editor, *Financial Cryptography and Data Security*, pages 6–24, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[57] Dorit Ron and Adi Shamir. How did dread pirate roberts acquire and protect his bitcoin wealth? In Rainer Böhme, Michael Brenner, Tyler Moore, and Matthew Smith, editors, *Financial Cryptography and Data Security*, pages 3–15, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[58] Tim Ruffing, Pedro Moreno-Sanchez, and Aniket Kate. Coinshuffle: Practical decentralized coin mixing for bitcoin. In Mirosław Kutyłowski and Jaideep Vaidya, editors, *Computer Security - ESORICS 2014*, pages 345–364, Cham, 2014. Springer International Publishing.

[59] Nicolas Van Saberhagen. Cryptonote v 2.0, 2013. whitepaper, Available online: https://cryptonote.org/whitepaper.pdf, (Last accessed 23-May-2019).

[60] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474, 5 2014.

[61] Michele Spagnuolo, Federico Maggi, and Stefano Zanero. Bitiodine: Extracting intelligence from the bitcoin network. In Nicolas Christin and Reihaneh Safavi-Naini, editors, *Financial Cryptography and Data Security*, pages 457–468, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[62] Surae Noether, Sarang Noether and A. Mackenzie. A Note on Chain Reactions in Traceability in CryptoNote 2.0, 9 2014. Available Online: https://ww.getmonero.org/resources/research-lab/pubs/MRL-0001.pdf (Last accessed 01-Jan.-2019).

[63] Muoi Tran, Loi Luu, Min Suk Kang, Iddo Bentov, and Prateek Saxena. Obscuro: A bitcoin mixer using trusted execution environments. In *Proceedings of the 34th Annual Computer Security Applications Conference*, ACSAC '18, pages 692–701, New York, NY, USA, 2018. ACM.

[64] F. Tschorsch and B. Scheuermann. Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys Tutorials*, 18(3):2084–2123, 2016.

[65] Luke Valenta and Brendan Rowan. Blindcoin: Blinded, accountable mixes for bitcoin. In Michael Brenner, Nicolas Christin, Benjamin Johnson, and Kurt Rohloff, editors, *Financial Cryptography and Data Security*, pages 112–126, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[66] D. A. Wijaya, J. Liu, R. Steinfeld, and D. Liu. Monero ring attack: Recreating zero mixin transaction effect. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 1196–1201, 8 2018.

[67] Dimaz Ankaa Wijaya, Joseph Liu, Ron Steinfeld, Dongxi Liu, and Tsz Hon Yuen. Anonymity reduction attacks to monero. In Fuchun Guo, Xinyi Huang, and Moti Yung, editors, *Information Security and Cryptology*, pages 86–100, Cham, 2019. Springer International Publishing.

[68] Dimaz Ankaa Wijaya, Joseph K. Liu, Ron Steinfeld, Dongxi Liu, and Jiangshan Yu. On the unforkability of monero. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, Asia CCS '19, pages 621–632, New York, NY, USA, 2019. ACM.

[69] Bitcoin Wiki. Technical background of version 1 bitcoin addresses. Available Online : https://en.bitcoin.it/wiki/Technical_background_of_version_1_Bitcoin_addresses/(Last accessed 15-Oct.-2018).

[70] P. Wuille. Bitcoin commit 5400ef6, 2015. Available Online: https://github.com/bitcoin/bitcoin/commit/5400ef6bcb9d243b2b21697775aa6491115420f3 (Last accessed 28-Jan-2019).

[71] J. Yu, M. H. A. Au, and P. Esteves-Verissimo. Re-thinking untraceability in the cryptonote-style blockchain. In *2019 IEEE 32nd Computer Security Foundations Symposium (CSF)*, pages 94–9413, June 2019.

[72] Zuoxia Yu, Man Ho Au, Jiangshan Yu, Rupeng Yang, Qiuliang Xu, and Wang Fat Lau. New empirical traceability analysis of cryptonote-style blockchains. In *Financial Cryptography and Data Security*, 2019.

[73] Jan Henrik Ziegeldorf, Fred Grossmann, Martin Henze, Nicolas Inden, and Klaus Wehrle. Coinparty: Secure multi-party mixing of bitcoins. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, CODASPY '15, pages 75–86, New York, NY, USA, 2015. ACM.

[74] Jan Henrik Ziegeldorf, Roman Matzutt, Martin Henze, Fred Grossmann, and Klaus Wehrle. Secure and anonymous decentralized bitcoin mixing. *Future Generation Computer Systems*, 80:448 – 466, 2018.