A Survey on Security and Privacy Issues of Bitcoin

Mauro Conti, *Senior Member, IEEE,* Sandeep Kumar E, *Member, IEEE,* Chhagan Lal, *Member, IEEE,* Sushmita Ruj, *Senior Member, IEEE*

***Abstract*—Bitcoin is a popular "cryptocurrency" that records all transactions in a distributed append-only public ledger called "blockchain". The security of Bitcoin heavily relies on the incentive-compatible distributed consensus protocol, which is run by participants called "miners". In exchange for the incentive, the miners are expected to honestly maintain the blockchain. Since its launch in 2009, Bitcoin economy has grown at an enormous rate, and it is now worth about 40 billions of dollars. This exponential growth in the market value of Bitcoin motivates adversaries to exploit weaknesses for profit, and researchers to identify vulnerabilities in the system, propose countermeasures, and predict upcoming trends.**

In this paper, we present a systematic survey on security and privacy aspects of Bitcoin. We start by presenting an overview of the Bitcoin protocol and discuss its major components with their functionality and interactions. We review the existing vulnerabilities in Bitcoin which leads to the execution of various security threats in the Bitcoin system. We discuss the feasibility and robustness of the state-of-the-art security solutions. We present privacy and anonymity considerations and discuss the threats to enabling user privacy, along with the analysis of existing privacy-preserving solutions. Finally, we summarize the critical open challenges and suggest directions for future research towards provisioning stringent security and privacy techniques for Bitcoin.

***Index Terms*—Bitcoins, cryptocurrency, security threats, user privacy**

I. Introduction

Digital transactions and online trading are gaining a lot of interest in e-commerce society. In such electronic payment systems, the consensus is reached via a trusted centralized authority that may appear as a bank, a Chartered Accountant (CA), a notary, or any other trusted service. The use of such third party authorities as an authenticator increases the cost of trading because a nominal fee is deducted as a payment or commission by these third parties. In 2008, a new concept called "Bitcoins" was introduced [1] that avoids this excessive cost caused by the transaction fee. Bitcoin is a cryptographically secure decentralized peer-to-peer (P2P) electronic payment system, and it enables transactions involving virtual currency in the form of digital tokens. Such digital tokens, also called Bitcoin Coins[1] (BTCs) are cryptocurrencies whose implementation relies on cryptography techniques. The cryptography is used in order to control the generation of new coins and to securely validate the transactions without involving any central authorities. In Bitcoin, the trust in a third-party such as a bank is replaced by a cryptographic Proof-of-Work (PoW) scheme that uses a public digital append-only ledger called *blockchain*. This ledger keeps records for all coin balances and transactions in the whole Bitcoin system that are announced, agreed upon, and that is completed in the past. The blockchain is accessible to all the network nodes (or participants) in order to enforce transparency in the system.

In [2], authors claim that "*Bitcoin works in practice and not in theory*" due to lack of the security research to find out theoretical foundation for Bitcoin protocols. Until today, due to the incomplete existence of robust theoretical base, security research community was dismissing the use of Bitcoin. Existing security solutions in Bitcoin lacks the required measures that could ensure an adequate level of security to its users. We believe that security solutions should target all the major protocols running critical functions in the Bitcoin system. These include blockchain protocol, peer-to-peer communication protocols, cryptographic protocols, and key management protocols. However, online communities have already started to adapt the Bitcoin, as it is believed that it will soon take over the online trading business. For instance, "Wiki leaks" request its users to donate using the coins. The request quote is "*Bitcoin is a secure and anonymous digital currency. Bitcoins cannot be easily tracked back to you, and are safer, and are the faster alternative to other donation methods*". Wiki leaks also support the use of Litecoin, another cryptocurrency, for the same reason [3].

Recently, Bitcoin technology is grabbing a lot of attention from government bodies. This is due to its increasing use by the malicious users to undermine legal controls. In [4], authors call Bitcoins "*Enigmatic and Controversial Digital Cryptocurrency*" due to mysterious concepts underneath the Bitcoin system and severe opposition from the government. According to [5], the current Bitcoin exchange rate is approximately 2000 dollars from around 600 dollars in mid-2016. The major technologies such blockchain and consensus protocols that makes the Bitcoin systems a huge success will be now envisioned in various next generation applications, which includes smart trading in smart grids [6], Internet of Things (IoT) [7] [8], vehicular networks [9], healthcare data

Prof. Mauro Conti, is with Department of Mathematics, University of Padua, Padua, Italy. e-mail:conti@math.unipd.it. The work of M. Conti was supported by a Marie Curie Fellowship funded by the European Commission under the agreement PCIG11-GA-2012-321980. This work is also partially supported by the EU TagItSmart! Project H2020-ICT30-2015-688061, the EU-India REACH Project ICI+/2014/342-896, the TENACE PRIN Project 20103P34XC funded by the Italian MIUR, and by the projects Tackling Mobile Malware with Innovative Machine Learning Techniques, Physical-Layer Security for Wireless Communication, and Content Centric Networking: Security and Privacy Issues funded by the University of Padua

Mr. Sandeep Kumar E, is with Department of Telecommunication Engineering, Ramaiah Institute of Technology, Bengaluru, India. e-mail:sandeepe31@gmail.com

Dr. Chhagan Lal, is with Department of Mathematics, University of Padua, Padua, Italy. e-mail:chhagan@math.unipd.it

Prof. Sushmita Ruj, is with Cryptology and Security Research Unit, Computer and Communication Sciences Division, Indian Statistical Institute, India. e-mail:sush@isical.ac.in

[1] In rest of the paper, we will use the terms *coin* and *BTCs* interchangeably.

management, and smart cities [10] [11], to name a few. As the length of popularity largely depends on the amount of security built on the system which surpasses all its other benefits, we aim to investigate the associated security and privacy issues in Bitcoin systems.

A. Contribution

In this paper, we present a comprehensive survey specifically targeting the security and privacy aspects in Bitcoin systems. We discuss the state-of-the-art attack vector which includes various user security and transaction anonymity threats that limits (or threatens) the applicability (or continuity) of Bitcoins in real-world applications and services. We also discuss the efficiency of various security solutions that are proposed over the years to address the security and privacy challenges in Bitcoin system. In particular, we mainly focus on the security challenges and their countermeasures with respect to major components of Bitcoin system that includes transaction, Blockchain, mining pools, and Bitcoin's networking protocols. In addition, we discuss the issues of user privacy and transaction anonymity along with a large array of research that has been done recently for enabling privacy and improving anonymity in Bitcoins.

In literature, authors in [12], provides a comprehensive technical survey on decentralized digital currencies with mainly emphasizing on Bitcoins. The authors explore the technical background of Bitcoin system and discuss the implications of the central design decisions for Bitcoin protocols. In [2], authors discuss the cryptocurrencies and Bitcoin in detail, and also provides a preliminary overview of the Bitcoin's payment systems pros and cons. However, the paper lacks a detailed survey about security and privacy attacks and their associated solutions. In particular, the main contributions that this survey provides are as follow.

- We present the required background knowledge for Bitcoin, its functionalities, and related concepts.
- We cover all the existing security and privacy-related threats that are associated with different components of Bitcoin system at various levels of its overall operation.
- We discuss the efficiency and limitations of the state-of-the-art solutions that address the security threats and enables strong privacy in Bitcoin systems, thus providing a holistic technical perspective on these issues in Bitcoin.

By doing so, we aim to assist interested readers in understanding existing security and privacy-related challenges, estimate the possible damage caused by these, and to improve the techniques for detection and containment of identified existing and future attacks in Bitcoins.

B. Organization

The rest of the paper is organized as follow. In Section II, we present a brief overview of Bitcoin which includes the description of its major components along with their functionalities and interactions. In Section III, we discuss a number of security threats associated with the development, implementation, and use of Bitcoin systems. In Section IV, we discuss the state-of-the-art proposals that either countermeasure a security threat or enhances the existing security in Bitcoins. In Section V, we discuss the anonymity and privacy threats in Bitcoins along with their existing solutions. We point out the potential lessons learned from our survey, and the future research directions towards the enhancement of security and privacy in today's Bitcoin system in Section VI. Finally, we conclude the paper in Section VII.

II. Overview of Bitcoin

Bitcoin is a decentralized electronic payment system introduced by Nakamoto [1]. It is based on peer-to-peer network and a probabilistic distributed consensus protocol. In Bitcoin, electronic payments are done by generating transactions that transfer *coins* among Bitcoin users. The source and destination addresses are represented by a cryptographic hash of a public key of the respective user. A user can have multiple addresses by generating multiple public keys and these addresses are associated be one or more of her wallets [13], but the private key of the user is required to spend coins in form of the digitally signed transactions. Using the hash of the public key as a receiving or sending address provides Bitcoin users a certain degree of anonymity, and it is recommended the practice to use different public keys for each transaction. In a Bitcoin system, there are three major components, namely: users (or customers), miners, management staff, Bitcoin exchanges, and wallets. Figure 1 shows the main functions and means to achieve those functions for all these components.

A transaction to transfer the coins consists of a set of inputs and outputs, and it has a unique identifier. Each output depicts the amount sent and the script program[2], whereas each input specifies a pointer to a previous transaction's outputs and a corresponding signature (e.g., the redeem script) that satisfies all the required spending requirements, i.e., one cannot spend more Bitcoins than specified in the inputs. Transactions are processed to verify their integrity, authenticity, and correctness by a group of resourceful Bitcoin network nodes called "Miners". In particular, instead of mining a single transaction, the miners bundle a number of transactions that are waiting for the network to get processed in a single unit called "block". The miner advertises a block to the rest of the Bitcoin network as soon as it completes its processing (or validation) in order to claim the mining rewards. This block is then verified by the majority of miners in the network before it is successfully added in the globally-readable distributed Bitcoin public ledger called "blockchain". The miner who mines a block receives a reward or incentive when the mined block is successfully added to the blockchain. We now present an overview of the major technical components and operational features that are essential for the practical realization of the Bitcoin systems.

A. Transaction and Proof-of-Work

Bitcoin uses transactions to move coins from one user wallet to another. In particular, the coins are represented in the form

[2] The script programs used in Bitcoin are (i) the "Pay-to-PubKeyHash" that requires a signature corresponding to an address, and (ii) the "Pay-to-script hash" that also enables multi-signature addresses by requiring a threshold of m signatures from n public keys.

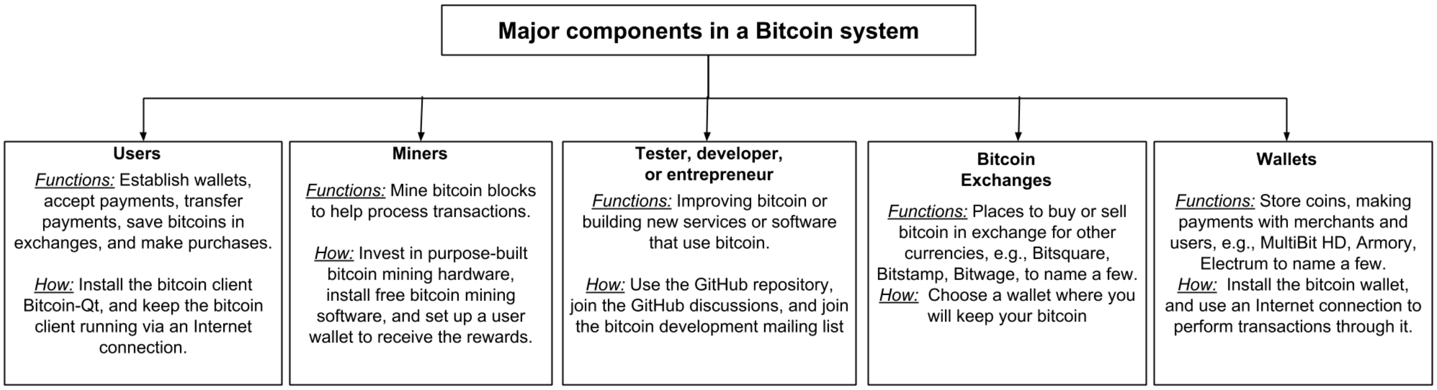
Major components in a Bitcoin system
Users
Functions: Establish wallets, accept payments, transfer payments, save bitcoins in exchanges, and make purchases.
How: Install the bitcoin client Bitcoin-Qt, and keep the bitcoin client running via an Internet connection.
Miners
Functions: Mine bitcoin blocks to help process transactions.
How: Invest in purpose-built bitcoin mining hardware, install free bitcoin mining software, and set up a user wallet to receive the rewards.
Tester, developer, or entrepreneur
Functions: Improving bitcoin or building new services or software that use bitcoin.
How: Use the GitHub repository, join the GitHub discussions, and join the bitcoin development mailing list
Bitcoin Exchanges
Functions: Places to buy or sell bitcoin in exchange for other currencies, e.g., Bitsquare, Bitstamp, Bitwage, to name a few.
How: Choose a wallet where you will keep your bitcoin
Wallets
Functions: Store coins, making payments with merchants and users, e.g., MultiBit HD, Armory, Electrum to name a few.
How: Install the bitcoin wallet, and use an Internet connection to perform transactions through it.

Fig. 1. Major components of a Bitcoin payment system

of transactions, more specifically, a chain of transactions. The key values in a transaction are one or more inputs, one or more outputs, and a unique transaction identifier (Tx_{id}) as depicted in Figure 2, where Tx_n is the n_{th} transaction in the blockchain. Briefly, each input belongs to a particular user, and it specifies the unspent coins, the hash of its previous transaction, and an index to one of its output. To authorize a transaction input, the corresponding user of the input provides the public key and the signature which is generated using her private key. As an input specifies the total number of unspent coins of a user, in each transaction the user has to operate on all of its remaining coins. For instance, Bob has 50 coins and he wants to transfer 5 coins to $Alice$. For this transaction, Bob has to make two different inputs, one showing a transaction in which 5 coins are transferred to Alice, and another showing a transfer of 45 coins in one (or more) wallet(s) owned by Bob. With this approach, the Bitcoin achieves two goals: (i) it implements the idea of $change$, and (ii) one can easily identify the unspent coins or balance of a user by only looking the outputs of its previous transaction. An output in a transaction specifies the number of coins being transferred along with the Bitcoin address of the new owner. These inputs and outputs are managed using a Forth-like scripting language which dictates the essential conditions to claim the coins. The dominant script in today's market is the "Pay-to-PubKeyHash" (P2PKH) which requires only one signature from the owner to authorize a payment, while another script such as the "Pay-to-ScriptHash" (P2SH) [14] enables a variety of transaction types and it supports future developments.

Unlike central bank in which all the transactions are verified, processed, and recorded in a centralized private ledger, in Bitcoin every user acts as a bank and keep a copy of this ledger. In Bitcoin, the role of the distributed ledger is played by the so-called blockchain. However, storing multiple copies of the blockchain in the Bitcoin network adds new vulnerabilities in the system such as keeping the global view of the blockchain consistent. For instance, a user (say $Alice$) could generate two different transactions simultaneously using the same set of coins to two different receivers (say, Bob and $Carol$). This type of malicious behavior by a Bitcoin user is termed as a *double spending*. If both the receiver processes the transaction independently based on their local view of the blockchain, and the transaction verification is successful, this leaves the blockchain into an inconsistent state. The main requirement to avoid the above problem is two-folded: (i) distribute the transaction verification process to ensure the correctness of the transaction, and (ii) everyone in the network should know quickly about a successfully processed transaction to ensure the consistent state of the blockchain. To fulfill the aforementioned requirements, Bitcoin uses the concept of *Proof-of-Work* (PoW) and a probabilistic distributed *consensus protocol*.

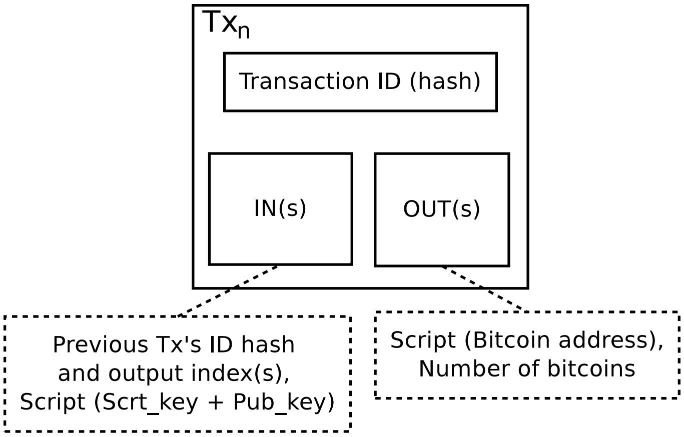
Tx$_n$
Transaction ID (hash)
IN(s)
OUT(s)
Previous Tx's ID hash and output index(s), Script (Scrt_key + Pub_key)
Script (Bitcoin address), Number of bitcoins

Fig. 2. Bitcoin transactions

The distributed transaction verification process ensures that a majority of miners will verify the legitimacy of a transaction before it is added in the blockchain. In this way, whenever the blockchain goes into an inconsistent state, all the nodes update their local copy of blockchain with the state on which a majority of miners agree, thus the correct state of the blockchain is obtained by election. However, this scheme is vulnerable to the sybil attacks [15]. With sybil attack, a miner creates multiple virtual nodes in the Bitcoin network and these nodes could disrupt the election process by injecting false information in the network such as voting positive for a faulty transaction. Bitcoin counters the sybil attacks by making use of PoW in which to verify a transaction, the miners have to perform some sort of computational task to prove that they are not virtual entities. The PoW consists of a complex cryptographic math puzzle [16], and it imposes a high level of computational cost on the transaction verification process, thus the verification will be dependent on the computing power of a miner, instead of the number of (possibly virtual) identities. The main idea is that it is much harder to fake the computing resources in the Bitcoin network than it is to perform a sybil attack.

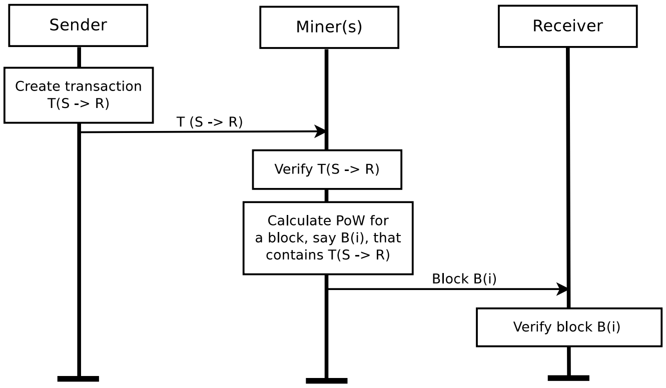
Sender
Miner(s)
Receiver
Create transaction T(S -> R)
T (S -> R)
Verify T(S -> R)
Calculate PoW for a block, say B(i), that contains T(S -> R)
Block B(i)
Verify block B(i)

Fig. 3. Bitcoin transaction execution process: A high-level view

In practical, the miners do not verify individual transactions, instead they collect pending transactions to form a *block*. The miners validate a block by calculating the hash of that block and vary a nonce value until the hash value becomes lower or equal to the given *target* value. Calculating the desired hash value is computationally difficult. Bitcoin uses the SHA-256 hash function [17]. Unless the cryptographic hash function finds the required hash value, the only option is to try different nonces until a solution (a hash value lower than target value) is discovered. Consequently, the difficulty of the puzzle depends on the target value, i.e., lower the target, the fewer solutions exist, the more difficult the hash calculation becomes. Once a miner calculates such a hash value for a block, it immediately broadcast the block in the network along with the calculated hash value, and it also appends the block in the public ledger (i.e., blockchain). The rest of the miners when receiving a mined block can quickly verify its correctness by comparing the hash value given in the received block with the *target* value. The miners will also update their local blockchain by adding the newly mined block. Once a block is successfully added in the blockchain (i.e., a majority of miners consider the block valid), the miner who first solved the PoW will be rewarded with a set of newly generated coins[3] and a small transaction fee [18]. Figure 3 depicts a high-level view of the Bitcoin transaction execution process, which starts from a transaction creation step and it ends when a block containing this transaction is mined successfully by miners residing in the Bitcoin network.

All the miner's race to calculate the hash value for a block by performing the PoW, so that they can collect the corresponding reward. The chance of being the first to solve the puzzle is higher for the miners who owns or controls more number of computing resources. By this rule, a miner with higher computing resources can always increase her chances to win the reward. To enforce stability, fairness, and reasonable waiting times for block validation, the $target$ value is adjusted after every 2,016 blocks. This adjustment of the $target$ also helps to keep per block verification time to approximately 10 minutes. It further effects the new coins generation rate in the Bitcoin network because keeping the block verification time near to 10 minutes implies that only 12.5 new coins can be added to the network per 10 minutes. In [19], authors propose an equation to calculate the new $target$ value for the Bitcoin system. The new target is given by the following Equation.

$$T = T_{prev} * \frac{T_{actual}}{2016 * 10min}. \tag{1}$$

Here, T_{prev} is the old $target$ value, and T_{actual} is the time period that the Bitcoin network took to generate the last 2,016 blocks.

B. Blockchain and Mining

The $blockchain$ is a public append-only link-list based data structure which stores the entire network's transaction history in terms of $blocks$ that combines the transactions in a Merkle Tree [20]. Along with each block, a relatively secure time-stamp and the hash of the previous block is also stored. Figure 4 shows the working methodology that has been used for creating and maintaining the blockchain in Bitcoin. To successfully add a new block in the blockchain, the miners need to verify (or mine) a block by solving a computationally difficult PoW puzzle. One can traverse the blockchain in order to determine the ownership of each coin because the blocks are stored in an ordered form. Also, tempering within a block is not possible as it would change the hash of the block. In particular, if a transaction in a block is tampered with, the hash value of that block changes, this, in turn, changes the subsequent blocks. The blockchain constantly grows in length due to the continuous mining process in the network. The process of adding a new block is as follows: (i) once a miner determines a valid hash value (i.e., a hash equal or lower than target) for a block, it adds the block in her local blockchain and broadcast the solution, and (ii) upon receiving a solution for a valid block, the miners will quickly check for its validity, if the solution is correct the miners update their local copy of blockchain else discard the block.

Due to the distributed nature of the block validation process, it is possible that two valid solutions are found approximately at the same time or distribution of a verified block is delayed due to network latency, hence it creates valid blockchain $forks$ of equal lengths. The forks are undesirable as the miners need to keep a global state of the blockchain that is consisting of the totally ordered set of transactions. However, when multiple forks exist, the miners are free to choose a fork and continue to mine on top of it. Now that the network is having multiple forks and miners are extending different but valid versions of the blockchain based on their local view, a time will come due to the random nature of PoW where miners of operating (or extending) one fork will broadcast a valid block before the others. Thus, a longer version of the blockchain now exists in the network, and due to the

[3] Currently the amount is 12.5 Bitcoin Coins (BTCs), and this reward is cut to half in every four years.

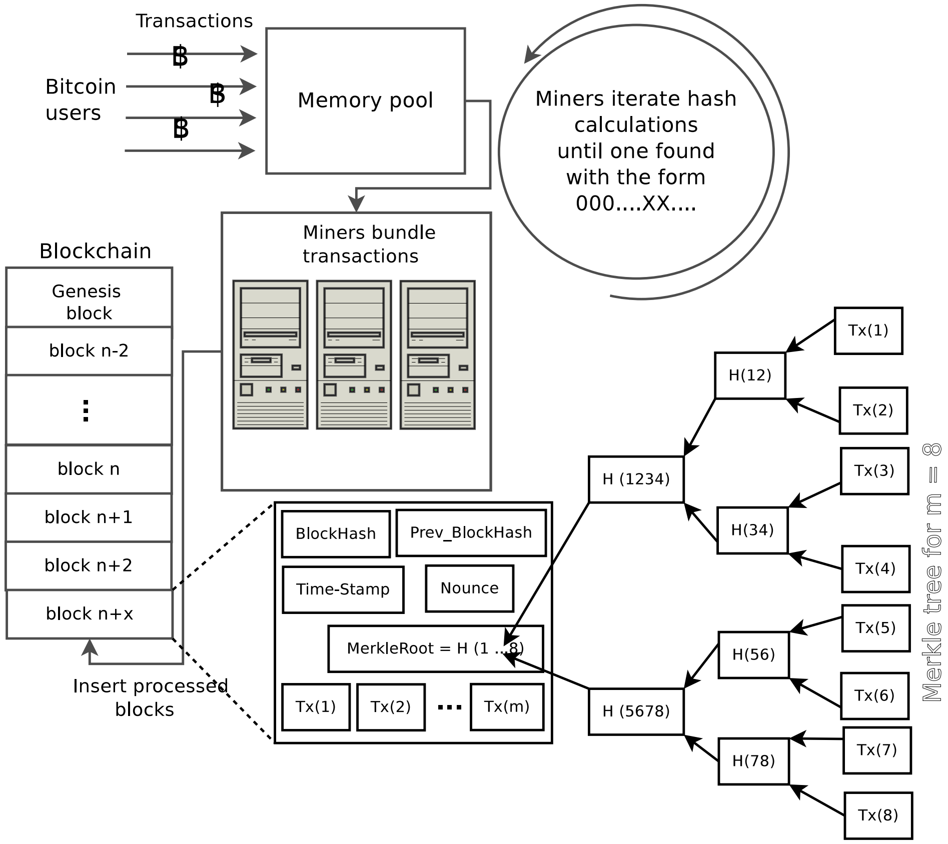

Fig. 4. Storage of blocks in blockchain

blockchain's consensus protocol, all the miners will start adding their next blocks on top of it. The aforementioned blockchain forking nature of Bitcoin could be exploited by a malicious miner to gain profits or to disturb the normal functioning of the Bitcoin systems. In particular, a resourceful miner (or mining pool) could force a blockchain fork in the Bitcoin network by privately mining on it to increase its length. Once the malicious miner sees that the length of the public blockchain is catching up fast with her private chain, the miner broadcast her chain into the Bitcoin network, and due to its longer length, all the other miners have to mine on top of it. In this process, all the mined (i.e., valid) blocks on the other parallel blockchain get discarded which makes the efforts of the genuine miners useless. In Section III, we will discuss an array of attacks on Bitcoin systems that are launched using the blockchain's forking feature. In general, the security in Bitcoin systems is on the assumption that the honest players control a majority of the computing resources.

The main driving factor for miners to honestly verify a block is the reward (i.e., 12.5 BTCs) that they receive upon every successful addition of a block in the global blockchain. As mentioned before that to verify a block, the miners need to solve the associated hard crypto-puzzle. The probability to solving the crypto-puzzle is proportional to a number of computing resources used. As per [21], a single home miner which uses a dedicated Application-Specific Integrated Circuit (ASIC) for mining will unlikely verify a single block in years. For this reason, miners mine in the form of the so-called *mining pools*. All miners that are associated with a pool works collectively to mine a particular block under the control of a pool manager. Upon a successful mining, the manager distributes the reward between all the associated miners proportional to the resources expended by each miner. A detailed discussion of different pooled mining approaches and their reward systems is given in [22] [23].

Finally, for better understanding the overall methodology of Bitcoin payment system, please refer to Figure 5. Assume that Bob wants to transfer 50 coins to $Alice$. In order to pay to $Alice$, Bob needs a device such as a smartphone, tablet, or laptop that runs the Bitcoin's client-side software, and two pieces of information which include $Bob's$ private key and $Alice's$ public key (also called as a Bitcoin address). Any user in the Bitcoin network can send money to a Bitcoin address, but only a unique signature generated using her own private key can release coins from her account. When Bob creates and broadcast a transaction in the Bitcoin network, an alert is sent to all the miners in the network, informing them about this new transaction. The miners verify that Bob has sufficient funds in order to complete the transaction by traversing into the blockchain (i.e., by checking its previous transaction outputs). Miners race to bundle all the pending transactions (including $bob's$) in the Bitcoin network and begin the block verification process by varying the nonce. The required hash value must have a certain but arbitrary number of zeros at the beginning. It is unpredictable which nonce has a correct number of zeros, so the miners have to keep trying by using different nonces

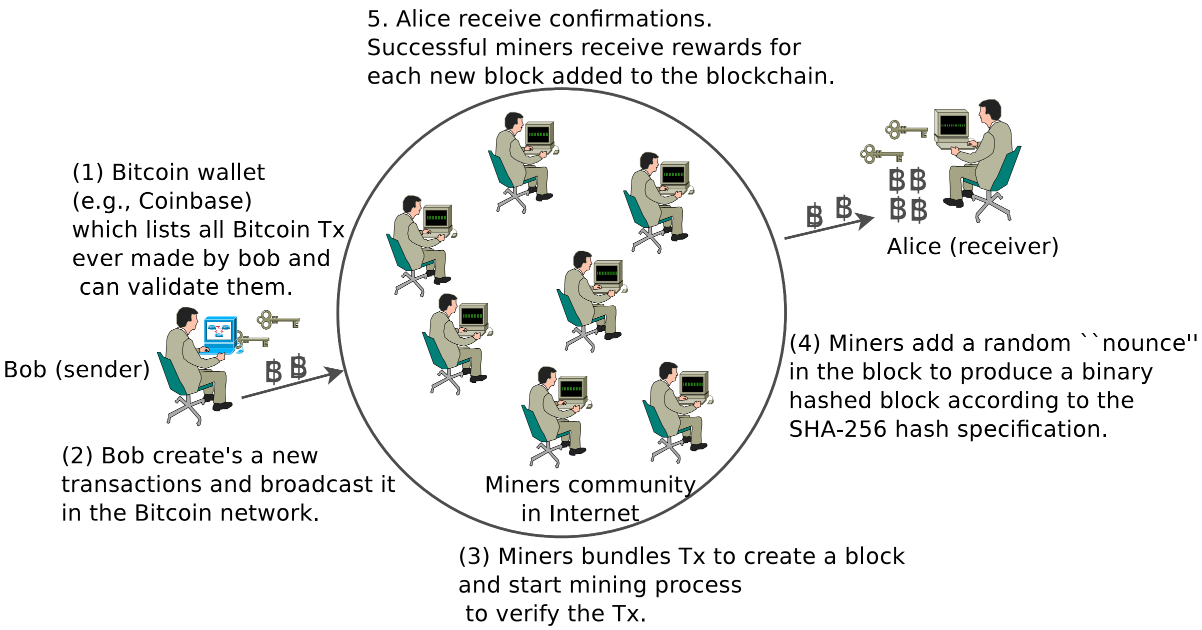
5. Alice receive confirmations.
Successful miners receive rewards for
each new block added to the blockchain.
(1) Bitcoin wallet
(e.g., Coinbase)
which lists all Bitcoin Tx
ever made by bob and
can validate them.
Bob (sender)
Alice (receiver)
(4) Miners add a random ``nounce''
in the block to produce a binary
hashed block according to the
SHA-256 hash specification.
(2) Bob create's a new
transactions and broadcast it
in the Bitcoin network.
Miners community
in Internet
(3) Miners bundles Tx to create a block
and start mining process
to verify the Tx.

Fig. 5. Bitcoin transaction process

to find the right value. When the miner finds a hash value with the correct number of zeros (i.e., the discovered value is lower than *target* value), the discovery is announced in the network, and both the *Bob* and the *Alice* will also receive a confirmation about the successful transaction[4]. Other miners communicate their acceptance, and they turn their attention to finding the next hash value for the next block of non-verified transactions.

The Bitcoin protocol rewards the winning miner with the set of newly created coins as *incentive*, and the hashed block is published in the public ledger. Within 10 minutes[5] of *Bob* initiating the transaction, he and *Alice* each receive the first confirmation that the Bitcoin was signed over to her. In terms of transaction time, the worst case is where the users have to wait for 10 minutes for the first confirmation as the mining process might involve the first time miners, else the time would be less. However, receiving the first confirmation does not mean that the transaction is processed successfully, and it cannot be invalidated at latter point of time. In particular, it has been recommended by the Bitcoin community that after a block is mined it should receive enough consecutive block confirmations (currently 6 confirmations) before it is considered as a valid transaction. This means on average it takes around one hour to safely assume that a transaction is validated successfully.

C. Advantages and Disadvantages

Like any other emerging technology, use of Bitcoin comes with certain pros and cons, and various types of risks are associated with its use. It is believed[6] that Bitcoin has the following pros and cons.

Pros:

- no intermediate organization can manipulate the currency or can have a hold on the transactions since every currency transfer happens peer-to-peer just like hard cash.
- anonymity and privacy are the major strengths of these kinds of virtual currencies. Transacting peers are pseudonymous since the transaction is via digitally signed coins which looks like a sequence of characters to an outsider.
- promotes a global economy that works everywhere, anytime, and with minimal processing fees.

Cons:

- the Bitcoin's mining process is governed by a crypto-puzzle. It is a strength. However, it consumes computing resources and require time (approximately 10 mins) before confirming a transaction.
- since there is no trusted third party like a bank, if password of crucial credentials are lost the user completely looses access to his account. Additionally, any crime and illegal transactions will possibly go unnoticed,
- Bitcoin transactions are irreversible, i.e., no refunds unless the receiver starts a new transaction to send the coins back to the sender.
- the use of Bitcoin encourages illicit activity such as money laundering, tax evasion, and illicit trade.

According to [24], the risk is the exposure to the level of danger associated with Bitcoin technology; in fact the same can be applied to any such digital cryptocurrency. The major

[4] Such a successful transaction could be discarded or deemed invalid at latter period of time if it is unable to stay in the blockchain due to reasons such as, existence of multiple forks, majority of miners does not agree to consider the block containing this transaction a valid block, a double spending attack is detected, to name a few.

[5] https://data.bitcoinity.org/bitcoin/block_time/5y?f=m10&t=l

[6] As some of these pros and cons are not entirely true at all the times, for instance Bitcoin transactions are not fully anonymous and the privacy of Bitcoin users could be threatened.

risks that threatens the wide usability of the Bitcoin payment systems are as follow:

- *Social risks:* it includes bubble formation (i.e., risk of socio-economic relationship such as what people talk and gossip), cool factor (i.e., entering the networking without knowing the ill effects), construction of chain (i.e., risk related with the blockchain formation like hashing and mining rewards), and new coins release (i.e., on what basis the new coins to be generated, is there a need etc.).
- *Legal risks:* Bitcoin technology opposes rules and regulations, and hence it finds opposition from the government. This risk also includes law enforcement towards handling financial, operational, customer protection and security breaches that arise due to Bitcoin system.
- *Economic risks:* deflation, volatility and timing issues in finding a block which might lead the users to migrate towards other currencies that offer faster services.
- *Technological risks:* this includes the following, network equipment, and its loss, network with which the peers are connected and its associated parameters, threat vulnerabilities on the system, hash functions with its associated robustness factor, and software associated risks that Bitcoin system demands.
- *Security risks:* security is a major issue in Bitcoin system, we will discuss risks associated due to various security threats in detail in Section III.

In [25], authors perform a survey on the people's opinion about Bitcoin usage. Participants argue that the greatest barrier to the usage of Bitcoin is the lack of support by higher authorities (i.e., government). Participants felt that Bitcoin must be accepted as legitimate and reputable currency. Additionally, the participants expressed that the system must provide support towards transacting fearlessly without criminal exploitation. Participants further state that the Bitcoin is mainly dependent on the socio-technical actors, and the impact of their opinion on the public. Few among participants have suggested that the blockchain construction is the major cause of disruption due to its tendency to get manipulated by adversaries.

In [26], it was stated that many Bitcoin users already lost their money due to poor usability of key management and security breaches, such as malicious exchanges and wallets. Around 22.5% of the participants reported having lost their coins due to security breaches. Also, many participants stated that for a fast flow of Bitcoins in the user community, simple and impressive user interface are even more important than security. In addition, participants highlighted that the poor usability and lack of knowledge regarding the Bitcoin usage are the major contributors for the security failures.

III. Security: Attacks on Bitcoin Systems

Bitcoin is the most popular cryptocurrency[7] and has stood first in the market capital investment from day one. Since it is a decentralized model with an uncontrollable environment, hackers and thieves find cryptocurrency system an easy way to fraud the transactions. In this section, we discuss existing security threats and their countermeasures for Bitcoin system. We provide a detailed discussion of potential vulnerabilities that can be found in the Bitcoin protocols as well as in the Bitcoin network, this will be done by taking a close look at the broad attack vector, and their impact on the particular components of the Bitcoin system. Apart from double spending, which is and will always be possible in Bitcoin, the attack space includes a range of wallet attacks (i.e., client-side security), network attacks (such as DDoS sybil and eclipse) and mining attacks (such as 50%, block withholding, and bribery). Table I provides a comprehensive overview of the potential security threats along with their impacts on various entities involved in the Bitcoin system and their possible solutions that exist in the literature so far.

A. Double Spending

A client in the Bitcoin network achieves a double spend (i.e., send two conflicting transactions in rapid succession) if she is able to simultaneously spend the same set of coins in two different transactions [27]. For instance, a dishonest client (C_d) creates a transaction $T_V^{C_d}$ at time t using a set of coins (B_c) with a recipient address of a vendor (V) to purchase some product from V. C_d broadcast $T_V^{C_d}$ in the Bitcoin network. At time t' where $t' \approx t$, C_d create and broadcast another transaction $T_{C_d}^{C_d}$ using the same coins (i.e., B_c) with the recipient address of C_d or a wallet which is under the control of C_d. In the above scenario, the double spending attack performed by C_d is successful, if C_d tricks the V to accept $T_V^{C_d}$ (i.e., V deliver the purchased products to C_d) but V will not be able to redeem subsequently. In Bitcoin, the *network of miners* verify and process all the transactions, and they ensure that only the unspent coins that are specified in previous transaction outputs can be used as input for a follow-up transaction. This rule is enforced dynamically at run-time to protect against the possible double spending in Bitcoin networks. The distributed time-stamping and consensus protocol is used for orderly storage of the transactions in the blockchain. For example, when a miner receives $T_V^{C_d}$ and $T_{C_d}^{C_d}$ transactions, it will be able to identify that both the transactions are trying to use the same inputs during the transaction propagation and mining, thus it only process one of the transaction and reject the other. Figure 6 shows the working methodology of a double spending attack depicting the above explanation.

Despite the use of strict ordering of transactions in the blockchain, proof-of-work scheme, distributed time-stamping [70], and consensus protocol [71] [72], double spending is still possible in Bitcoin network. To perform a successful double spending attack, following requirements need to be fulfilled: (i) part of the Bitcoin miners network accepts the transaction $T_V^{C_d}$ and the vendor (V) receives the confirmation from the miners, thus releases the product to dishonest client (C_d), (ii) at the same time, part of the Bitcoin miners network accepts the transaction $T_{C_d}^{C_d}$, thus create blockchain forks in the network, (iii) the vendor receives the confirmation of transaction $T_{C_d}^{C_d}$ after accepting the transaction $T_V^{C_d}$, thus losses the product, and (iv) a majority of miners mine on top of the blockchain which contains $T_{C_d}^{C_d}$ as a valid

[7] www.cryptocoinsnews.com/

TABLE I
BITCOIN ATTACKS, ADVERSE EFFECTS, AND COUNTERMEASURES

Attack	**Description**	**Primary targets**	**Adverse effects**	**Possible countermeasures**
Double spending or Race attack [27]	spent the same coins in multiple transactions, send two conflicting transactions in rapid succession	sellers or merchants	sellers lose their products, drive away the honest users from network, creates blockchain forks	inserting observers in the network [27], communicating double spending alerts among peers [27], nearby peers should notify the merchant about an ongoing double spend as soon as possible [28], merchants should disable the incoming connections [29] [30]
Finney attack [31]	dishonest miner broadcasts a pre-mined block for the purpose of double spending as soon as it receives product from a merchant	sellers or merchants	facilitates double spending	merchants should wait for multi-confirmation messages for a transaction
Brute force attack [32]	privately mining a long blockchain fork to perform double spending	sellers or merchants	facilitates double spending and creates large size blockchain forks	inserting observers in the network [27], notify the merchant about an ongoing double spend as soon as possible [29]
Vector 76 or one-confirmation attack [33]	combination of the double spending and the finney attack	Bitcoin exchange services	facilitates double spending with larger number of coins	merchants should wait for multi-confirmation messages for a transaction
$>50\%$ *hashpower or Goldfinger* [34]	adversary controls more than $>50\%$ of computational power in the Bitcoin network	Bitcoin network, miners, Bitcoin exchange centers, and users	drive away the miners working alone or within small mining pools, weakens the effectiveness of consensus protocol, DoS	inserting observers in the network [27], communicating double spending alerts among peers [27], disincentivize large mining pools [35] [36], TwinsCoin [37], PieceWork [38]
Block discarding [39] [30] or Selfish mining [40]	miner (or mining pool) withhold the processed block(s) in order to earn inappropriate incentives	honest miners (or mining pools)	introduce race conditions by forking, waste the computational power of honest miners, with $>50\%$ it leads to Goldfinger attack	ZeroBlock technique [41] [42], timestamp based techniques such as freshness preferred [43], DECOR+ protocol [44]
Block withholding [21] [45]	pool member withholds an already mined block	honest miners (or mining pools)	waste resources of fellow miners and decreases the pool revenue	include only known and trusted miners in pool, dissolve and close a pool as soon as the revenue drops from expected [39], cryptographic commitment schemes [45]
Bribery attacks [46]	adversary pay money to miners to mine on her behalf	miners and merchants	increases the success probability of carrying out a double spending or block withholding attack	increase the rewards for honest miners, communicate the miners that bribery might cause the long-term losses to the miners (including the dishonest miner) [46]
Refund attacks [47]	adversary exploits the refund policies of existing payment processors	sellers or merchants, users	merchant losses money while honest users might lose their reputation	use publicly verifiable evidence [47]
Punitive and Feather forking [48] [49]	dishonest miners want to blacklist transactions from a specific address	users	freeze the money held by Bitcoin users for forever	remains an open challenge
Transaction malleability [50] [51]	adversary can change the TXID without invalidating the transaction	Bitcoin exchange centers	Bitcoin exchange losses fund due to increase in double deposit or double withdrawal instances	use multiple metrics for transaction verification along with TXID [52], malleability-resilient “refund” transaction [50]
Wallet theft [13]	adversary stole or destroy private key of users	individual users or businesses	all the money in the wallet is lost	use of threshold signatures to achieve two-factor security [53] [54], use of hardware wallets [55], TrustZone-backed Bitcoin wallet [56]
Time jacking [57]	adversary speed-up the majority of miner’s clock	miners	isolate a miner and waste its computational resources, influence the mining difficulty calculation process	use constraints tolerance ranges [57], network time protocol (NTP) or time sampling on the values received from trusted peers [58]
Sybil [15]	adversary creates multiple virtual identities in the network	Bitcoin network, miners, users	facilitates time jacking, DoS, and double spending attacks, threatens user privacy	Xim (a two-party mixing protocol) [59]
DDoS [60] [61]	adversary exhaust the network resources by launching a collaborative attack	Bitcoin network, businesses, miners, and users	deny services to honest users/miners, isolate or drive away the miners	Proof-of-Activity (PoA) protocol [62], stronger authentication with fast verification signatures
Eclipse or netsplit [63]	adversary monopolizes all of the victim’s incoming and outgoing connections	miners and users	inconsistent view of the network/block chain at the attacked node, enable double spends with more than one confirmation	use whitelists, disabling incoming connections [63]
Tampering [64]	delay the propagation of transactions and blocks to specific nodes	miners, users	mount DoS attacks, considerably increase its mining advantage in the network, double spend transactions	modification of the block request management system [64]
Deanonymization [65] [66]	linking IP addresses with a Bitcoin wallet or public key address	users	user privacy violation/leakage	mixing services [67], CoinJoin [68], CoinShuffle [69]

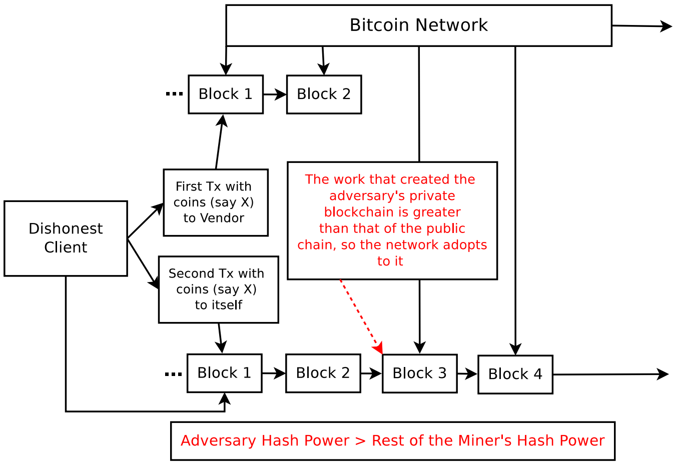
Bitcoin Network
...
Block 1
Block 2
First Tx with coins (say X) to Vendor
The work that created the adversary's private blockchain is greater than that of the public chain, so the network adopts to it
Dishonest Client
Second Tx with coins (say X) to itself
...
Block 1
Block 2
Block 3
Block 4
Adversary Hash Power > Rest of the Miner's Hash Power

Fig. 6. Double Spending Attack

transaction. If the aforementioned steps took place in the given order then the dishonest client is able to perform a successful double spend in the Bitcoin network. In the rest of this section, we will discuss the variants of double spending attack that are used in order to realize the aforementioned double spend requirements with varying difficulties and complexities.

A form of double spending called *Finney attack* [31], here a dishonest client (C_d) pre-mines (i.e., privately) a block which contains the transaction $T_{C_d}^{C_d}$, and then it creates a transaction $T_V^{C_d}$ using the same coins for a vendor (V). The mined block is not informed to the network, and the C_d waits until the transaction $T_V^{C_d}$ is accepted by the V. On the other hand, V only accept $T_V^{C_d}$ when it receives a confirmation from miners indicating that $T_V^{C_d}$ is valid and included in the existing blockchain. Once C_d receives the product from V, the attacker releases the pre-mined block into the network, thus creates a blockchain fork (say B'_{fork}) of equal length to the existing fork (say B_{fork}). Now, if the next mined block in the network extends B'_{fork} blockchain instead of B_{fork}, than as per the Bitcoin protocol rules all the miners in the network will build on top of B'_{fork}. As the blockchain B'_{fork} becomes the longest chain in the network, all the miners ignore B_{fork}, thus the top block on B_{fork} which contains the transaction $T_V^{C_d}$ becomes invalid. This makes the transaction $T_V^{C_d}$ invalid, the client will get back her coins through transaction $T_{C_d}^{C_d}$, but resulting the V losing the product. With *Finney attack* an adversary can only perform double spending in the presence of one-confirmation vendors.

To avoid the *Finney attack*, the vendor should wait for multiple confirmations before releasing the product to the client. The waiting for multiple confirmations will only make the double spend for the attacker harder, but the possibility of the double spend remains. An advancement of the *Finney attack* is called *Brute-force attack* [32] in which a resourceful attacker has control over n nodes in the network, and these nodes collectively work on a private mining scheme with the motive of double spend. An attacker introduces a double spend transaction in a block as in the previous case, while continuously works on the extension of a private blockchain (i.e., B'_{fork}). Suppose a vendor waits for x confirmations before accepting a transaction, and it sends the product to the client once it receives the x confirmations. Later, the the attacker is able to mine the x number of blocks ahead (i.e., privately) then she can release these blocks in the network, and due to its higher length than B_{fork}, blockchain B'_{fork} will be extended by all the miners in the network. This causes the same after effects as *Finney attack*, thus causing a successful double spending attack.

Another attack that uses the privately mined block to perform a new form of double spending attack on Bitcoin exchange networks is popularly known as *Vector 76 attack* [33]. In this attack, a dishonest client (C_d) withholds a pre-mined block which consists of a transaction that implements a specific deposit (i.e., deposit coins in a Bitcoin exchange). The attacker (i.e., C_d) waits for the next block announcement and quickly sends the pre-mined block along with the recently mined block directly to the Bitcoin exchange or towards its nearby peers with hope that the exchange and probably some of the nearby miners will consider the blockchain containing the pre-mined block (i..e, B'_{fork}) as the main chain. The attacker quickly sends another transaction that requests a withdrawal from the exchange for the same coins that was deposited by the attacker in its previous transaction. At this point of time, if the other fork (i.e., B_{fork}) which does not contain the transaction that is used by the attacker to deposit the coins survives, the deposit will become invalidated but the attacker has already performed a withdrawal by now, thus the exchanges losses the coins. Recently, authors in [73] proposes a new attack against the PoW mechanism in blockchain systems called the *Balance attack*. The attack consists of delaying network communications between multiple subgroups of miners with balanced hash power. The theoretical analysis provides the precise trade-off between the Bitcoin network communication delay and the mining power of the attacker(s) needed to double spend in Ethereum [74] with high probability.

Based on the above discussion on double spending attack and its variants, one main point that emerges is that if a miner (or mining pool) is able to mine blocks with a faster rate than the rest of the Bitcoin network, the possibility of a successful double spending attack is high. The rate of mining a block depends upon solving the associated proof-of-work, this again depends on the computing power of a Bitcoin node. Apart from the computing resources, the success of double spending attack depends on other factors as well which includes network propagation delay, vendor, client and Bitcoin exchange services connectivity or positioning in the Bitcoin network, and the number of honest miners. Clearly, as the number of confirmations for transaction increases, the possibility that it will become invalid at a later stage decreases, thus decreases the possibility of a double spend. On the other hand, with the increase in the computing resources (or the so-called *hash power*) of a miner, the probability of the success of a double spend increases. This leads to a variant of double spend attack called $> 50\%$ *attack* or *Goldfinger attack* [34] in which more than 50% computing resources of the network are under the control of a single miner. The $> 50\%$ *attack* is considered the worst-case scenario in the Bitcoin network because it has the power to destroy the stability of the whole network by introducing the actions such as claim all the block intensives, perform double spending, reject or include transactions as preferred, and play with the Bitcoin exchange rates. The instability in the network, once started, it will further strengths the attacker's position as more and more honest miners will start leaving the Bitcoin network.

From the above discussion on different type of double spending attacks, we can safely conclude that one can always perform a double spend or it is not possible to entirely eliminate the risk of double spending in Bitcoin. However, performing double spending comes with a certain level of risk, for instance, the attacker might lose the reward for the withheld block if it is not included in the final public blockchain. Therefore, it is necessary to set a lower bound on the number of double spend coins, and this number should compensate the risk of unsuccessful attempts of double spend. Additionally, the double spends could be recognized with the careful analysis and traversing of the blockchain, thus it might lead to blacklisting the detected peer. In Section IV-A, we will discuss in detail, the existing solutions and their effectiveness for detecting and preventing the double spending attacks.

B. Mining Pool Attacks

Mining pools are created in order to increase the computing or hash power which directly affects the verification time of a block, thus it increases the chances of wining the mining reward. For this purpose, in recent years, a large number of mining pools have been created, and the research in the field of miner strategies is also evolved. At the same time, the attack vector that exploits the vulnerabilities in pool based mining also increases. For instance, a group of dishonest miners could perform a set of internal and external attacks on a mining pool. Internal attacks are those in which miners act maliciously within the pool to collect more than their fair share of collective reward or disrupt the functionality of the pool to distant it from the successful mining attempts. In external attacks, miners could use their higher hash power to perform attacks such as double spending on the Bitcoin network. Figure 7 shows the market share till March 2017 of the most popular Bitcoin mining pools. In this section, we will discuss a set of popular internal and external attacks on the mining pools.

In [39], authors use a game theoretic approach to show that in the current Bitcoin payment system, the miners could have a specific sort of subversive mining strategy called *selfish mining* [40] or also popularly known as *block discarding attack* [39] [30]. In truth, all the miners in the Bitcoin are *selfish* as they are mining for the reward that is associated with each block, but these miners are also honest and fair with respect to the rest of the Bitcoin miners, while the *selfish mining* here refers to the malicious miners only. In the selfish mining attack, the dishonest miner(s) perform information hiding (i.e., withhold a mined block) as well as perform its revealing in a very selective way with a two-fold motive: (i) obtain an unfair reward which is bigger than their share of computing power spent, and (ii) confuse other miners and lead them to waste their resources in a wrong direction. By keeping the mined block(s), the selfish miners intentionally fork the blockchain. The selfish pool keeps on mining on top of their private chain (B'_{fork}), while the honest miners are mining on the public chain (B_{fork}). If the selfish miners are able to take a greater lead on B'_{fork} and they are able to keep the lead for a longer time period, their chances of gaining more reward coins as well as the wastage of honest miners resources increases. As soon as the B_{fork} reaches to the length of B'_{fork}, the selfish miners publish their mined blocks. All the miners need to adopt to B'_{fork} which now becomes B_{fork} as per the longest length rule of Bitcoin protocol. The honest miners will lose their rewards for the blocks that they have mined and added to the previous public chain. The analysis presented in [40], shows that using the selfish mining, the pool's reward exceed its share of the network's mining power. The wastage of computing resources and rewards lure honest miners toward the selfish mining pools, thus it further strengthens the attack. This continuous increase in the selfish pool's size might lead to $> 50\% attack$ at that point the effect of selfish mining will be disastrous.

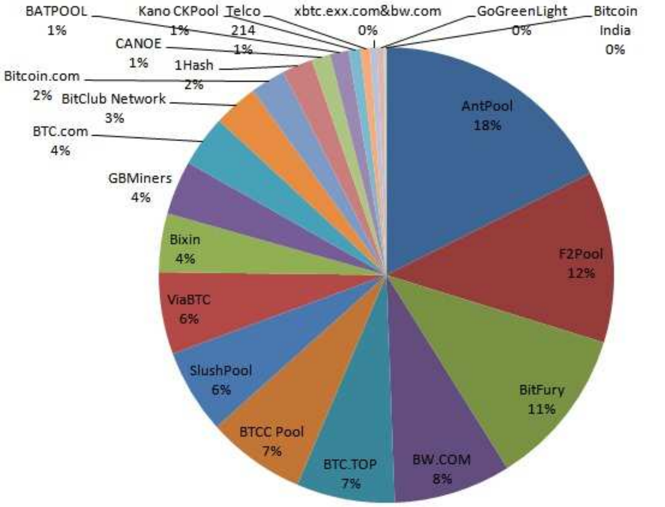

Fig. 7. Mining Power Distribution in Present Market

The *Pool Hopping attack* presented in [21] [75] can be considered as a type of selfish mining. In this attack, the adversary performs continuous analysis of the number of shares submitted by fellow miners to the pool manager in order to find (i.e., publish) a block. The idea is that if already a large number of shares have been submitted and no block has been found so far, the adversary will be getting a very small share from the reward because it will be distributed based on the shares submitted. Therefore, at some point in time, it might be more profitable for the adversary to switch to another pool or mine independently. Another attack much similar to the block discarding attack that could be performed on a mining pool by a malicious miner is known as *Block withholding* [21] [45], in which a pool member never publishes a mined block in order to sabotage the pool revenue. In [21], two type of block withholding scenarios are presented called "Sabotage" and "Lie in wait". In the first scenario, the adversary does not gain any coins, but it just makes other pool members loose, while in the second scenario, the adversary performs a complex block concealing attack similar to the one described in the *selfish mining* attack. In [21], authors discuss a generalized version of the "Sabotage" attack which shows that with slight modification, it is possible for the malicious miner to also earn an additional profit in this scenario. Authors in [76] present a game-theoretic approach to analyzing effects of block withholding attack on mining pools. The analysis shows that the attack is always well-incentivized in the long-run, but may not be so for a short duration. This implies that existing pool protocols are insecure, and if the attack is conducted systematically, Bitcoin pools could lose millions of dollars worth in just a few months.

Recently, the *Bribery attack* is described in [46]. In this attack, an attacker might obtain the majority of computing resources (i.e., mining capacity) for a short duration via bribery. Authors discuss three ways to introduce bribery in the network: (i) Out-of-Band Payment, in which the adversary pays directly to the owner of the computing resources and these owners then mine blocks assigned by the adversary, (ii) Negative-Fee Mining Pool, in which the attacker forms a pool by paying higher return, and (iii) In-Band Payment via Forking, the attacker attempts to bribe through Bitcoin itself by creating a fork containing bribe money freely available to any miner adopting the fork. By having the majority of the hash power, the attacker could launch different attacks in the Bitcoin such as double spending and Distributed Denial-of-Service (DDoS) [77]. The miners that took the bribes will get benefit which will be short-lived, but these short-lived benefits might be undermined by the losses in the long run due to the presence of DDoS and Goldfinger attacks or via an exchange rate crash in the network.

In [48], authors present a malicious mining strategy called *feather forking*, in which a dishonest miner attempts to blacklist one or more transactions from a client by publicly announcing not to extend a blockchain if it contains one of the blacklisted transaction, thus it will retaliate by forking the chain. The adversary forks as per its convenience, and it will continue to extend its fork until it wins (i.e., outraces the main chain), but if it is losing (i.e., falls behind as compared to the main chain by a predefined n blocks) than it discards its fork and continue to extend the main chain. An adversary with total hash power less than 50% might, with high probability, lose rewards, but it will be able to block the blacklisted transaction with positive probability. Moreover, if the adversary can show that he is determined to block the selected transaction and will perform the retaliatory forking if required, then the rest of the miners will also be motivated to block the blacklisted transactions to avoid the losses, in case, if the attacker retaliates and wins. If this is the case, an attacker might be able to enforce the selective blacklisting with no real cost because other miners are convinced that the attacker will perform a costly feather forking attack if provoked. An attacker performing *feather forking* can also use it to *blackmail* a client by threating that all her transactions will be put on the blacklist until the client pays the asked ransom coins.

C. Client-side Security Threats

The huge increase in the popularity of Bitcoin encouraged a large number of new users to join the network. Each Bitcoin client posses a set of private-public keys in order to access its Bitcoin account or wallet. Hence, it is desirable to have the key management techniques that are secure, yet usable. This is due to the fact that unlike many other applications of cryptography, if the keys of a client are lost or compromised, the client will suffer immediate and irrevocable monetary losses. To use Bitcoin, a user needs to install a wallet in her desktop or mobile device. The wallet stores the set of private-public keys associated with the owner of the wallet, thus it is essential to take protective actions to secure the wallet. The *wallet thefts* are mainly performed using mechanisms that include system hacking, installation of buggy software, and incorrect usage of the wallet.

Bitcoin protocol relies heavily on elliptic curve cryptography [78] for securing the transactions. In particular, Bitcoin uses elliptic curve digital signature algorithm (ECDSA) which is standardized by NIST [79] for signing the transactions. For instance, consider the standard "Pay-to-PubKeyHash" (P2PKH) transaction script in which the user needs to provide her public key and the signature (using her private key) to prove the ownership. To generate a signature, the user chooses a per-signature random value. For security reason this value must be kept secret, and it should be different for every other transaction. Repeating per-signature value risks the private key computation, as it has been shown in [80] that even partially bit-wise equal random values suffice to derive a user's private key. Therefore, it is essential for increasing the security of ECDSA to use highly random and distinct per-signature values for every transaction signature. The inspection of the blockchain for instances, in which the same public key uses the same signature nonces for multiple times has been reported by the authors in [81]. In particular, the authors report that there are 158 public keys which have reused the signature nonce in more than one transaction signature, thus making it possible to derive user's private keys. Recently, authors in [82] presents a systematic analysis of the effects of broken primitives on Bitcoin. Authors highlight the fact that in the current Bitcoin system no migration plans are in-place for both the broken hash and the broken signature scheme, i.e., the Bitcoins RIPEMD160, SHA256, and ECDSA techniques are vulnerable to various security threats such as collision attacks [83]. The authors in [82] found that the main vectors of attack on Bitcoin involve collisions on the main hash or attacking the signature scheme, which directly enables coin stealing. However, a break of the address hash has minimal impact, as addresses do not meaningfully protect the privacy of a user.

Unlike most of the online payment systems that rely on login details consisting of the password and other confidential details for user authentication, Bitcoin relies on public key cryptography. This arises the issues of the secure storage and management of the user keys. Over the years, various type of wallet implementations are researched to obtain secure storage of the user keys, it includes software, online or hosted,

TABLE II
BITCOIN WALLETS

	Coinbase	**Blockchain**	**TREZOR**	**Exodus**	**MyCelium**	**Bitcoin Core**	**MultiBit HD**	**Electrum**	**Copay**	**Armory**
Wallet type	Hot wallet	Hot wallet	Hardware wallet	Hot wallet	Hot wallet	Hot wallet	Hot wallet	Hot wallet	Multisig	Varies
Web interface	Yes	Yes	Yes	No	No	No	No	No	Yes	No
Mobile app	Yes	Yes	No	No	Yes	No	No	No	Yes	No
Desktop client	No	No	No	Yes	No	Yes	Yes	Yes	Yes	Yes
Independent wallet	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Privacy	Moderate	Weak	Variable	Good	Good	Good	Moderate	Good	Good	Good
Security	Good	Good	Good	Good	Good	Good	Good	Moderate	Good	Good/Moderate

hardware or offline, paper and brain wallets. Table II shows a number of popular wallets and their main features. Coinbase (coinbase.com), an online wallet is most popular due to its desirable features which it provides to the clients that include: (i) a web interface using which the wallet can be assessed with a browser and Internet connection, (ii) a mobile app that allows access to wallet through mobile devices, (iii) an access to Coinbase does not require a client software and it is independent in nature due to which the wallet providers does not have any control on the funds stored in a client's wallet, and (iv) a moderate level of security and privacy. The *Copay* wallet allows multiple users to be associated with the same wallet, while the *Armory* wallet works in online as well as in offline mode. The wallet providers have to find an adequate trade-off between usability and security while introducing a new wallet into the market. For instance, an online wallet is more susceptible to thefts compared to hardware wallets [55] as later are not connected to the Internet, but at the same time hardware wallets lacks usability. If done right, there exists more advanced and secure ways to store the user keys called *paper* and *brain* wallets. As their name indicates, in the paper wallet the keys are written on a document which is stored at some physical location analogizes the cash money storage system, while in brain wallet the keys are stored in the clients mind in the form of a small passphrase. The passphrase if memorized correctly is then used to generate the correct private key.

To avoid the aforementioned risks such as managing cryptographic keys [84], lost or stolen devices, equipment failure, Bitcoin-specific malware [85], to name a few, that are associated while storing the coins in a wallet, many users might prefer to keep their coins with online exchanges. However, storing the holdings with an exchange makes the users vulnerable to the exchange systems. For instance, one of the most notorious events in the Bitcoins history is the breakdown and ongoing bankruptcy of the oldest and largest exchange called Mt. Gox, which lost over 450M US dollars. Moreover, a few other exchanges have lost their customers' Bitcoin savings and declared bankruptcy due to external or internal theft, or technical mistakes [86]. Although, the vulnerability of an exchange system to the disastrous losses can never be fully avoided or mitigated, therefore the authors in [87] presents *Provisions*, which is a privacy-preserving proof of solvency for Bitcoin exchanges. Provision is a sensible safeguard that requires the periodic demonstrations from the exchanges to show that they control enough Bitcoins to settle all of its customers accounts.

D. Attacks on Bitcoin Protocols and Networking Infrastructure

In this section, we will discuss those attacks in the Bitcoin that exploits, the existing vulnerabilities in the implementation and design of the Bitcoin protocols and its peer-to-peer communication networking protocols. We will start our discussion with the most common networking attack called *Distributed Denial-of-Service* (DDoS) attack which targets Bitcoin currency exchanges, mining pools, eWallets, and other financial services in Bitcoin. DDoS attacks are inexpensive to carry out, yet quite disruptive in nature. In these attacks, the adversary exhausts the network resources in order to disrupt their access to genuine users. For example, a honest miner is congested with the requests (such as fake transactions) from a large number of clients acting under the control of an adversary. After a while, the miner will likely to start discarding all the incoming inputs/requests including requests from honest clients. In [60], authors provide a comprehensive empirical analysis of DDoS attacks in the Bitcoin by documenting the following main facts: 142 unique DDoS attacks on 40 Bitcoin services and 7% of all known operators were victims of these attacks. The paper also states that the majority of DDoS attack targets the exchange services and large mining pools because a successful attack on these will earn huge revenue for the adversary as compared to attacking an individual or small mining pools.

In [61], authors explore the trade-off between the two mining pool related strategies using a series of game-theoretical models. The first strategy called *construction*, in which a mining pool invests for increasing its mining capacity in order to increase the likelihood of winning the next race. While in the second strategy called *destruction*, in which the mining pool launches a costly DDoS attack to lower the expected success rate of a competing mining pool. The majority of the DDoS attacks target large organizations due to bulk ransom motives. Companies like CoinWallet and BitQuick were forced to shutdown only after few months of their launch due to the effects of continuous DoS attacks. As stated above that DoS attack take various forms, one of which discourages a miner so that it will withdraw itself from the mining pool. For instance, an attacker displays to a colleague miner that it is more powerful, and it can snatch the reward of mining, and it is the obvious winner of the mining process. An honest miner backoffs since its chances of winning is less. Hence, an adversary is successful in removing individual miners as well as small pools from the mining network, thus imposing a DoS attack in the network [61]. Moreover, in [88], authors propose network partitioning in Bitcoin peer-to-peer networks, thus isolating the honest nodes from the network by reducing their reputation.

Now we discuss the so-called *Malleability attacks* [51], which also facilitates the DDoS attacks in Bitcoin networks. For instance, by using a *Malleability attack* an adversary clogs the transaction queue [89]. This queue consists of all the pending transactions which are about to be serviced in the network. Meanwhile, an adversary puts in bogus transactions with the high priority depicting itself to be highest incentive payer for the miners. When the miners try to verify these transactions, they will find that these are the false transaction, and but by this time they have already spent a considerable amount of time in verifying these false transactions. Hence, this attack wastes the time and resources of the miners and the network [90]. Malleability is defined in terms of cryptography by [51]. A cryptographic primitive is considered malleable, if its output Y can be "mauled" to some "similar" value Y' by an adversary who is unaware of the cryptographic secrets that were used to develop Y.

In [50], another form of *malleability* attack called *transaction malleability* is introduced. In Bitcoin, suppose that a transaction $T^n_{A \to B}$ which transfers n coins from $A's$ wallet to $B's$ wallet, with *transaction malleability* it is possible to create another T' that is syntactically different (i.e., $T^n_{A \to B}$ and T' has different transaction hash ID T^{id}_x) from $T^n_{A \to B}$, although semantically it is identical (i.e. T' also transfers n coins from wallet A to B). An adversary can perform *transaction malleability* without even knowing the private key of A. On a high level, *transaction malleability* refers to a bug in the original Bitcoin protocol which allows the aforementioned behavior in the network possible. The main reason is that, in Bitcoin each transaction is uniquely identified by its T^{id}_x, and hence in some cases T' will be considered a different transaction than $T^n_{A \to B}$.

In Bitcoin, certainly, the transaction malleability is not desirable, but it does not cause any damage to the system until an adversary exploits its behavior and make someone believe that a transaction has been failed. However, after a while the same transaction gets published in the global blockchain. This might lead to a possible double spend, but it is particularly more relevant while targeting the Bitcoin exchanges which holds a significant amount of coins. This is because it allow the Bitcoin users to buy and sell coins in exchange of cash money or altcoins. The Bitcoins reference implementation is immune to the transaction malleability because it uses previous transaction's outputs as an indication for the successfully issued transactions. However, few exchanges use a custom implementation and were apparently vulnerable. For instance, Mt. Gox (a popular exchange) issued a statement in the early days of Bitcoin that they were attacked due to transaction malleability, therefore they are forced to halt withdrawals and freezing clients account. The attack that MtGox claimed to be the victim proceeds as follows: (i) an dishonest client C_d deposits n coins in his MtGox account, (ii) C_d sends a transaction T to MtGox asking to transfer her n coins back, (iii) MtGox issues a transaction T' which transfers n coins to C_d, (iv) C_d performs the malleability attack, obtaining T' that is semantically equivalent to T but has a different T^{id}_x, now assume that T' gets included into the blockchain instead of T, (v) C_d complains to MtGox that the transaction T was not successful, (vi) MtGox performs an internal check, and it will not found a successful transaction with the T^{id}_x, thus MtGox credits the money back to the user's wallet. Hence effectively C_d is able to withdraw her coins twice. The whole problem is in the above Step (vi), where MtGox should have searched not for the transaction with T^{id}_x of T, but for any transaction semantically equivalent to T.

Due to the vulnerabilities that exist in the refund policies of the current Bitcoin payment protocol, a malicious user can perform the so-called *Refund attacks*. In [47], authors present the successful implementation of the refund attacks on $BIP70$ payment protocol. BIP70 is a Bitcoin community-accepted standard payment protocol that governs how vendors and customers perform payments in Bitcoin. Most of the major wallets use BIP70 for coin exchanges, and the two dominant Payment Processors called *Coinbase* and *BitPay*, who uses BIP70 and collectively they provide the infrastructure for accepting coins as a form of payment to more than 100,000 vendors. The authors propose two types of refund attacks called *Silkroad Trader attack* which highlights an authentication vulnerability in the BIP70, and *Marketplace Trader attack* which exploits the refund policies of existing payment processors. The brief description of both the refund attacks is as follows:

- In the *Silkroad attack*, a customer is under the control of an ill trader. When a customer starts trading with the merchant, its address is revealed to the ill trader. When the transaction is finished, the adversary initiates the attack by inserting the customer's address as the refund address and send a refund request to the merchant. The merchant sends the amount to the ill merchant and hence gets cheated without receiving a refund from the other side. During this whole process of refund between the merchant and the ill trader, the customer is not at all aware of the fraud that is happening in her name.
- The *Marketplace trader attack* is a typical case of the man-in-the-middle attack. In this, the adversary setup an attractive webpage, where she attracts the customer who falls victim in the later stages. The attacker depicts himself as a trusted party by making payments through trustable merchants like CeX. When a customer clicks the webpage, accidentally she reveals her address among the other identities that are sufficient to perform malpractice by the rogue trader with the false webpage. When customer purchase products, a payment page is sent which is a legitimate payment exchange merchant. The end merchant is connected to the adversary's webpage and meanwhile, the details of the customer would have been already revealed to the attacker, through an external email communication according to the Bitcoin refund policies. After the transaction, the middle adversary claims a refund on behalf of the customer, and the refund amount will be sent to the rogue adversary's account. Hence, the legitimate customer will not be aware of the fraud process, but the merchant loses his coins [47].

Later, both these attacks have been acknowledged by Coinbase and Bitpay with temporary mitigation measures put in place. However, the authors claim that to fully address the identified issues will require revising the BIP70 standard.

Yet another attack on the Bitcoin networks is called *Time jacking attack* [57]. In Bitcoin network, all the participating nodes internally maintain a time counter that represents the network time. The value of the time counter is based on the median time of a node's peers, and it is sent in the version message when peers first connect. However, if the median time differs by more than 70 minutes from the system time, the network time counter reverts to the system time. An adversary could plant multiple fake peers in the network, and all these peers will report inaccurate timestamps, thus it can potentially slow down or speed up a node's network time counter. An advanced form of this attack would involve speeding up the clocks of a majority of the network's mining resources while slowing down the target's clock. Since the time value can be skewed by at most 70 minutes, the difference between the nodes time would be 140 minutes [57]. Furthermore, by announcing inaccurate timestamps, an attacker can alter a node's network time counter and deceive it into accepting an alternate blockchain because the creation of new blocks heavily depends on network time counters. This attack significantly increases the possibility of the following misbehaviors: a successful double spending attack, exhaust computational resources of miners, and slow down the transaction confirmation rate.

Apart from the aforementioned major attacks on Bitcoin protocol and network, their are few other minor attacks that we are summarized below.

- *Sybil Attack:* A type of attack, where attacker installs dummy helper nodes and tries to compromise a part of the Bitcoin network. A sybil attack [15] is a collaborative attack performed by a group of compromised nodes. Also, an attacker may change its identity and may launch a collusion attack with the helper nodes. An attacker tries to isolate the user and disconnect the transactions initiated by the user, or a user will be made to choose only those blocks that are governed by the attacker. If no nodes in the network confirm a transaction, that input can be used for double spending attack. An intruder with her helper nodes can perform a collaborated timing attack, thus can hamper a low latency encryption associated with Bitcoin network. The other version of this attack where the attacker tries to track back the nodes and wallets involved in the transaction is discussed in [59].
- *Eclipse attack:* In this attack [63], an adversary manipulates a victim peer. The IP addresses to which the victim user connects are blocked or diverted towards an adversary [63]. In addition, an attacker can hold multiple IP addresses to spoof the victims from the network. An attacker may deploy helpers and launch other attacks in the network such as double spending and selfish mining. The attack could be of two type: (i) Infrastructure attacks, where attack is on the ISP (Internet Service Provider) which holds numerous contiguous addresses, hence it can manipulate multiple addresses that connect peer-to-peer in the network, and (ii) botnet attacks, where an adversary can manipulate addresses in a particular range, especially in small companies which own their private set of IP addresses. In both the cases, an adversary can manipulate the peers in the Bitcoin network.
- *Tampering:* In a Bitcoin network, after mining a block the miners broadcast the information about newly mined blocks. New transactions will be broadcast from time to time in the network. The network assumes that the messages will reach to the other nodes in the Bitcoin network with a good speed. However, authors in [64] ground this assumption, and they prove that the adversary can induce delays in the broadcast packets by introducing congestion in the network or making a victim node busy by sending requests to all its ports. Such type of tampering can become a root cause for other types of attacks in the network.

IV. Security: Countermeasures for Bitcoin Attacks

In this section, we discuss the state-of-the-art security solutions that provides possible countermeasure for the array of Bitcoin attacks presented in Section III.

A. No more double spending

Bitcoin's default solution against double spending is to use its *Proof-of-Work* (PoW) technique, which limits the capabilities of an adversary in terms of her computational resources. The concept of PoW also protects the network against being vulnerable to sybil attacks which, if launched it could sabotage the functionality of consensus algorithm and leads to possible double spending attack. In general, double spending could be dealt in two possibles ways: (i) detect a double spending instance by monitoring the blockchain progress, and once detected, identify the adversary and take adequate actions, or (ii) use preventive measures. The former approach works well in the traditional centralized online banking system, but in Bitcoin it's not suitable due to the use of continuously varying public keys as a wallet address, thus it provides anonymity to users, and the lack of transaction rollback scheme once it is successfully added in the blockchain. Therefore, the later approach, i.e., prevent double spend, is desirable in Bitcoin.

Authors in [27] evaluate three techniques that can be used to detect a possible double spending in fast payment systems namely: using a *listening period*, *inserting observers*, and *forwarding double spending attempts*. In the first technique, the vendor associates a listening period with each received transaction, and it monitors all the receiving transactions during this period. The vendor only delivers the product, if it does not see any attempt of double spending during its listening period. The *inserting observers* technique naturally extends the first technique based on the adoption of a listening period would be for the vendor to insert a set of nodes (i.e., "observers") under its control within the Bitcoin network. These observers will directly relay all the transactions to the vendor that they receive from the network. In this way, with the help of the observers, the vendor is able to *see* number of transactions in the network during its *listening period*, thus increases the chances of detecting a double spend. The third technique (i.e., *forwarding double spending attempts*) requires each Bitcoin peer to forward all transactions that attempt to double spend instead of discarding them so that the vendor can receive such a transactions on time (i.e., before releasing the product). With this approach, whenever a peer receives a new transaction, it checks whether the transaction is an attempt to double spend, if so, then peer forward the transaction to their neighbors (without adding it to their memory pools).

Recently, the hash power of a pool called *GHash.IO* reached 54% for a day (i.e., it exceeds the theoretical attack threshold of 51% in Bitcoin). Although the *GHash.IO* remained honest by transferring a part of its mining power to other pools. But the incentives that motivate an adversary to create large pools remains in the network, looking for a chance to wrongful gain and disrupt the network. Therefore, a method to prevent the formation of large pools called *Two phase Proof-of-Work* (2P-PoW) has been proposed in [36]. The authors propose a second proof-of-work (say Y) on top of the traditional proof-of-work (say X) of the block header. Y signs the produced header with the private key controlling the payout address. Similar to existing hashing procedures this signature must meet a target set by the network, thus the use of Y forces pool managers to distribute their private key to their clients if the manager wants to retain the same level of decentralization. However, if a manager would naively share its private key, all clients would be authorized to move funds from the payout address to any destination. Pool managers unwilling to share their private key, therefore, need to install mining equipment needed to solve Y in a timely manner. It is estimated that GHash.IO owns only a small percentage of the network's computing power in hardware, as the pool shrank significantly after public outrage. Depending on the difficulty $Y's$ cryptographic puzzle, this would only allow a certain number of untrusted individuals to join. This would, as GHash.IO is a public pool, severely limit its size.

Another solution to control double spending was proposed in [91] where all the participating users deposit a safety amount similar to an agreement. If an attacker tries to double spend and it is detected, the deposit amount will be deducted and given to the victim who encountered the loss. Due to the punishing attribute of the network, the attack can be controlled. In [30], authors suggest a countermeasure by prohibiting the merchant to accept incoming connections, thus an adversary cannot directly send a transaction to the merchant. This forces the adversary to broadcast the transaction over the Bitcoin network, and it ensures that the transaction will end up in the local view of all the miners that forwards it. Later if the adversary tries to double spend the miners will know about it.

Solution for 50% attack is presented in [30]. The authors provide countermeasures for two variants of 50% attack namely: *block discarding attack* and *difficulty rising attack*. In block discarding attack, an adversary has control over a set of nodes in the network, called *supporters*. The adversary and her supporters purposefully add delay to the legitimate block, and the attacker advertises her block selfishly. Hence, the advertiser's blockchain will increase, and the other blocks due to delay get less attention. The delay becomes worse as the number of supporter increases. The solution for this attack is fixing the punishment for the advertisers or the misbehaving miners. Every node is asked to pay a deposit amount, and the nodes who misbehave are punished by dissolving the deposit amount of the concerned. This amount is distributed among the nodes who informs about the misbehaving node in the network. In difficulty rising attack, the attacker manipulates the network and slowly raises the difficulty level for the miners. An attacker poses a threat to the network with high hash-power compared with other nodes in the network. The solution to this attack is same as that of block discarding attack. In [92], authors propose a method called "proof-of-reputation", where the honest miners will get a token based on the current market value. The number of tokens issued can vary with the market value. If the miner has the token, he will be reputed in the mining market pool. The token has a value, and according to which the coins are deposited from all the miners from time to time and is fixed by the network. More the reputation of the miner's chain, more the other blocks merge with that chain.

For now, it is safe to conclude that there is no solution available in the literature that guarantees the complete protection from double spending in Bitcoin. The existing solutions only make the launching of double spending attack more difficult for the adversary. In particular, double spending is an attack that is well discussed in the Bitcoin community, but very few solutions exist so far, and it remains an open challenge for the researchers. The easiest, yet most powerful way for a vendor to avoid a double spend, is to wait for more number of confirmations before accepting a transaction. Therefore, each vendor or merchant of the Bitcoin has to set a trade-off between the risk and the product delivery time caused while waiting for an appropriate number of confirmations. Similar to the honest Bitcoin users, there is also a trade-off for the adversary as she needs to consider the expenses (i.e., the loss of computing resources and rewards for the pre-mined blocks) if the attack fails.

B. *Countermeasures for Private Forking and Pool Attacks*

When a dishonest miner intentionally forks the blockchain by privately mining a set of blocks, it makes the Bitcoin network vulnerable to a wide range of attacks such as selfish mining, block-discarding attack, block withholding attack, bribery attacks to name a few. The aim of these attacks is to cheat Bitcoins mining incentive system. Therefore, at any point in time, detecting and mitigating the faulty forks from the set of available forks poses a major challenge for Bitcoin protocol developers. The simplest solution to handle the selfish mining is suggested in [40]. The authors propose a simple, backwards-compatible change to the Bitcoin protocol. In particular, when a miner encounters the presence of multiple forks of the same length, it will forward this information to all its peers, and it randomly chooses one fork to extend. Hence, each miner implementing the above approach by selecting a random fork to extend. This approach will decrease the selfish pool's ability to increase the probability that other miners will extend their fork.

To further extend the countermeasure presented in [40], authors in [43] introduce the concept of *Freshness Preferred*

(FP), which places the unforgeable timestamps in blocks and prefer blocks with recent timestamps. This approach uses Random Beacons [93] in order to stop miners from using timestamps from the future. As the selfish mining uses strategic block withholding technique, the proposed strategy will decrease the incentives for selfish mining because withheld blocks will lose block races against newly minted or "fresh" blocks. A similar, but a more robust solution for selfish mining that requires no changes in existing Bitcoin protocol is proposed in [41]. The authors suggest a fork-resolving policy that selectively neglects blocks that are not published in time, and it appreciates blocks that include a pointer to competing blocks of their predecessors. Therefore, if the secretly mined block is not published in the network until a competing block is published, it will contribute to neither or both branches, thus it gets no benefits in winning the fork race.

Unlike most of the aforementioned solutions against malicious forking, authors in [42] propose a timestamp-free prevention of block withholding attack called *ZeroBlock*. In ZeroBlock, if a selfish miner keeps a mined block private more than a specified interval called *mat*, than later when this block is published on the network it will be rejected by honest miners. The key idea is that each consecutive block must be published in the network, and it should be received by honest miners within a predefined maximum acceptable time for receiving a new block (i.e., *mat* interval). In particular, an honest miner either receives or publishes the next block in the network within the *mat* interval. Otherwise, to prevent the block withholding, the miner itself generates a specific block called *Zeroblock*. For forking attacks that are internal to a pool, authors in [39] suggest that the only viable option to countermeasure a block withholding attack launched within a pool is that, the pool managers should involve *ONLY* miners which are personally known to them, hence they can be trusted. The pool manager should simply dissolve and close a pool, as soon as the earning of the pool goes lower than expected from its computational effort.

In [46], bribery attack is discussed along with its countermeasure. In bribery, an attacker bribe a miner to rent her computing resources, thus it increases the attackers hash power that it could use to launch various attacks in Bitcoin networks. As a countermeasure, authors suggest the use of anti-payment (i.e, counter-bribing) to pool miners which have value more than what attackers are paying to these miners to perform a malicious behavior. However, the drawback is that a legitimate pool manager has to spend a lot to take miners toward the normal mining routine. In addition, as the number of bribing node or a node's bribe amount increases, the capital requirements for the manager also increases, and as the crypt math becomes more and more difficult the bribe amount increases, thus makes it difficult for the manager to keep the process of counter-bribing active for longer periods.

C. Securing Bitcoin wallets

A wallet contains private keys, one for each account [84]. These private keys are encrypted using the master key which is a random key, and it is encrypted using AES-256-CBC with a key derived from a passphrase using SHA-512 and OpenSSLs EVP_BytesToKey [94]. Private key combined with the public key generates a digital signature which is used to transact from peer-to-peer. Bitcoin uses ECDSA (Elliptic Curve Digital Signature Algorithm) algorithm for encryption, and it is modified in [81] for secret sharing and threshold cryptography.

A manual method of wallet protection was proposed by [95] called a "cold wallet". A cold wallet is another account that holds the excess of an amount by the user. This method uses two computers (the second computer has to be disconnected from the Internet) and using the Bitcoin wallet software a new private key is generated. The excess amount is sent to this new wallet using user's private key. Authors in [95] claim that if the computer is not connected to the Internet, the hackers will not get to know the keys, thus the wallet safety can be achieved. Securing wallets with new cryptographic algorithms apart from ECDSA is still an open issue and a challenge. In [96], an article states that US government have launched their own Bitcoin networks with multi-factor security which incorporates fingerprint biometrics for wallet protection. A device is a standalone tool same as the size of a credit card. In [55], authors propose *BlueWallet*, a proof-of-concept based hardware token for the authorization of transactions in order to protect the Bitcoin private keys. The concept is similar to the use of the "cold wallet", that is, it uses a dedicated hardware not connected to the Internet to store the private keys. The hardware token communicates with the computer (or any other device) that creates the transaction using Bluetooth Low Energy (BLE) and it can review the transaction before signing it. The securely stored private key never leaves the BlueWallet and is only unlocked if the user correctly enters her PIN. BlueWallet provides the desired security on the expense of the usability, as the users have to invest and keep an additional device while making a transaction.

Bitcoin already has a built-in function to increase the security of its wallets called "multi-signature", which tightens the security by employing the splitting control technique. For instance, *BitGo* - an online wallet provides 2-of-3 multi-signature transactions to its clients. However, the drawback of using the multi-signature transactions is that it greatly compromises the privacy and anonymity of the user. Authors in [53], proposes an efficient and optimal threshold Digital Signature Algorithm (DSA) scheme for securing Bitcoin keys. The main idea behind the use of threshold signatures proposed in [53] is derived from secret sharing [97], in which the private key is split into shares. Any subset of the shares that is equal to or greater than a predefined threshold is able to reconstruct the private key, but any subset that is smaller will gain no information about the key. The main property of threshold signatures [54] is that the key is never revealed because the participants directly construct a signature. Recently, authors in [56] present a TrustZone[8] based Bitcoin wallet and shows that it is more resilient to the dictionary and side-channel attacks. Although the use of TrustZone make use of the

[8] TrustZone is a technology that is used as an extension of processors and system architectures to increase their security.

encrypted storage, thus the writing and reading operations become slower.

D. Securing Bitcoin Protocol and Network

In this section, we will discuss various existing countermeasures proposed for securing the Bitcoin's core protocol stack and its peer-to-peer networking infrastructure functionalities against an array of security threats some of which we have presented in Section III-D.

1) DDoS Attacks: In [61], authors propose a game theoretic approach for analyzing the DDoS attacks. The game assumes that the pools are in competition with each other because the larger pools are always weighted more than the smaller pools. The game exists between the pools, and each pool tries to increase their computational cost over others, and then it imposes a DDoS attack on the other pools. Hence, authors draw an equilibrium condition between the players, and it concludes that the larger pools have more incentives against the smaller pools. In [98], authors propose a "miner's dilemma", again a game theoretical approach to the behavior of miners similar to repetitive prisoner's dilemma. There exist a game between the Bitcoin pools. The longest chain dominates over the smaller chains and grabs the rewards by behaving selfishly in the network. Game theory concludes that by performing attacks, the pools actually lose the Bitcoins that they are supposed to get when compared it with the case without attacking each other. In particular, this kind of game theory problems is called "Tragedy of Commons", where the peers turn out to be rational, selfish and harm other peers for their benefits.

In [62], author's proposes Proof-of-Activity (PoA) protocol, which is robust against a DoS attack that could be launched by broadcasting a large number of invalid blocks in the network. In PoA, each block header is stored with a crypt value and the user that stores the first transaction places this value. These users are called "stakeholders" in the network and they are assumed, to be honest. Any subsequent storage of transactions to this block is done if there are valid stakeholders associated with the block. Storage of crypt value is random, and more transactions are stored, only if more stake users are associated with the chain. If the length of the chain is more, trustworthiness among other peers increases and more miners get attracted towards the chain. Hence, an adversary cannot place a malicious block or a transaction, since all the nodes in the network are governed by stakeholders.

One possible way to mitigate DDoS attacks is to use the technique discussed in [99], which suggests the continuous monitoring of network traffic by using browsers like Tor or any user-defined web service. Applying machine-learning techniques like SVM and clustering will identify which part of the network is behaving ill. Hence, that part can be isolated from the Bitcoin network until debugged. Other possible methods to protect against DoS attacks include: (i) configure the network in a way that malicious packets and requests from unnecessary ports will be prohibited, (ii) implement a third party DoS protection scheme which carefully monitors the network and identify variations in the pattern. We believe that similar approaches could also be implemented in future in Bitcoin networks to countermeasure DoS attacks.

2) Time Jacking and Eclipse Attack: In this attack, an adversary alters the node time, therefore the dependency of a node on network time can be replaced by a hardware oriented system time. The accept time window (for transactions on a node) has to be reduced, making the node recover quicker from the attacks. *Time jacking* is a dreaded attack that might split the network into multiple parts and hence, it can isolate the victim node. A set of techniques is suggested in [57] avoid time jacking that includes, use system time instead of the network time to determine the upper limit of block timestamps, tighten the acceptable time ranges, and use only trusted peers. Even a node can be designed to hold multiple timestamps assuming that the attacker may not alter all the timestamps. Node timestamps can be made dependent on the blockchain timestamps [57].

In [63], authors provide techniques to combat eclipse attack which uses an additional procedure to store the IP addresses that are trustworthy. If the users are connected to another peer in the Bitcoin network, they are stored in "tried" variable. The connection of the user with the peers is dependent on the threshold of the trust factor, which varies from time to time. The users can have special intrusion detection system to check the misbehaving nodes in the network. The addresses which misbehave in the network are banned from connections. These features can prevent the Bitcoin user under eclipse by an attacker(s). In particular, having a check on the incoming and outgoing connections from the node can reduce the effect of an eclipse attack.

3) Refund Attacks and Transaction Malleability: In [47], modifications are proposed in the *Payment Request* message by adding information about the customer such as registered e-mail address, delivery address, and product information. The payment address should be unique for each Payment Request. Each request is associated with a key, and the same key is used for a refund, however, the use of the additional information might threaten the privacy of the customer. The customer is no longer involved in the information broadcast about the transaction, but the responsibility is to handover the refund to the merchant. Hence, all the nodes will learn about the transaction during verification and can identify the attacker easily. In particular, the idea is to provide the merchant, a set of publicly verifiable evidence which can cryptographically prove that the refund address received during the protocol belongs to the same pseudonymous customer who authorized the payment.

In [100], authors propose a manual intervention process that checks the withdrawal transactions to detect a possible malleability attack. Any suspicious pending transactions in the blocks can be seen as a sign of the attack. In addition, all the transactions on the Bitcoin network should have confirmations. In [50], authors show a case of malleability attack on "deposit protocol", and provide a solution namely *new deposit protocol*.

4) Reducing Delays in Transactions: In Bitcoin practice, the transactions with large Bitcoins are not usually carried out due to the risk of losing it or fear of fraudulent activities. Hence, the transaction is broken into a set of smaller transactions. But, eventually, it increases the delay in the network due to the time the network spends in validating the transactions. Hence to reduce this delay, authors in [101] suggest payments offline through a separate type of transactions called “micropayments” [102] and via a separate channel called micropayment channel. This channel is not a separate network but part of Bitcoin network itself. In a traditional Bitcoin network, users broadcast their transaction and the other miners verify it. This happens for all the transactions, and the network might get clogged at places where a large number of transactions exist. Also, in such a situation, the network gives preference to transactions with large denomination compared to the smaller ones. Hence, by establishing micropayment channels, the separate dedicated channel is allocated for the counterparties to perform the transaction. The basic idea is that the transaction is not revealed until both the parties trust each other on their balances and transaction that they perform. If either of the ones misbehaves, then the transaction is broadcasted for the verification from the Bitcoin network. The channels obey the Bitcoin protocol and they are established like any other naive network routing techniques. Hence, these micro payment channels constitute a “lightning network”. The advantages of using a lightning network are as follows:

- The technique provides high-speed payments, eliminates the dependency on the third party to validate, reduced load on the Bitcoin network, channels can stay open indefinitely for the transactions, counterparties can move out of the agreement whenever they want, parties can sign using multiple keys.
- Parties can broadcast their information when they want for seeking the interference of the other miners to solve the discrepancies.
- Parties can send their transaction over the channel without revealing their identities to the network and the nodes helping in routing.
- Payments a be routed across many block chains. The network allows micro level payment transactions.

5) Tampering: In [64], author’s provide solutions for *tampering attacks*. A node can announce the time it takes to mine a block together with the advertisement of a new block. This makes another peer in the network approximately estimate the average time needed to mine a block, and hence no one can spoof by adding unnecessary delays or tampering timestamps. Instead of static timeouts, dynamic timeouts can make more sense, since mining time can vary from node to node. All the senders buffer the IP addresses with which it is connecting every time, and this avoids the IP sending same advertise messages again and again to the same peer. A track of all the nodes has to be recorded in every sender and pattern can be analyzed. If a transaction is not replied by a node in a time window, then the sender will ask other nodes to confirm the transaction.

Despite all the security threats and their solutions that we have discussed, the number of miners in a network is a factor of consideration. More the miners, more people to verify the transactions, hence faster the block validation process, and more efficient the consensus process. However, the miners are incentive driven, hence the reward coins can pull more miners into the process, but at the same time the reward reduces half for every four years, thus the miners may migrate towards other cryptocurrencies which offer them more reward coins.

Bitcoin’s consensus algorithm has been its most widely debated component in the Bitcoin research community. This is because the consensus algorithm rises: (i) open questions about the Bitcoin stability [2]; (ii) concerns about the performance and scalability of the protocol [103]; and (iii) concerns that its computational puzzle wastes computational resources [76]. In particular, the blockchain protocol underlying the Bitcoin system is very inefficient in terms of power consumption and the overall generation time of new blocks, which is due to the associated PoW that is utilized to make the overall protocol work. Discussing the improvements or alternatives for Bitcoin’s consensus algorithm for addressing the above research issues are out of the scope of our survey, hence we direct the inserted users to read the current research trends regarding it in [104] [105] [2]. Similarly, for the same aforementioned reason, in this survey, we don’t discuss the alternatives to the proof-of-work such as proof-of-stake [106], proof-of-publication [107], proof-of-burn [108], proof-of-activity [62], to name a few, and for in-depth details, we direct interested users to [109] [12].

As the security issues in Bitcoin are closely linked with the user privacy and anonymity, we discuss the threats and their existing countermeasures for enabling privacy and enhancing anonymity for Bitcoin users in detail in the next section.

V. Privacy and Anonymity in Bitcoin

Bitcoin technology upholds itself when it comes to the privacy, but the only privacy that exists in Bitcoin comes from pseudonymous addresses (i.e., public keys) which are fragile and easily compromised through different techniques such as Bitcoin address reuse, “taint” analysis and tracking payments via blockchain analysis methods, IP address monitoring nodes, web-spidering, to name a few. Once broken, this privacy is difficult and sometimes costly to recover. Bitcoin allows its user to trivially generate new Bitcoin addresses (i.e., public keys) for each transaction, this provides a strong privacy as argued in [1]. In a traditional banking system, the transactions are known only by the bank and the involved parties, while in the Bitcoin system the public blockchain reveals all the transaction data to any user connected to the Bitcoin network. The original white paper on Bitcoin claims that this reveal of transactions information through blockchain does not disclose the identity of the parties involved in these transactions, and in [65] authors clarify that the Bitcoin system does not have any directory to maintain the log and other transaction related information. However, an adversary can associate the offline data such as emails and shipping addresses with the online information, and it can get the private information about the peers. In particular, Bitcoin offers a partial unlinkability (i.e., pseudonymity), and thus it is possible to link a number of transactions to an individual Bitcoin user by tracing the flow of money through a robust blockchain analysis procedure. In this section, we discuss the various security threats to privacy and

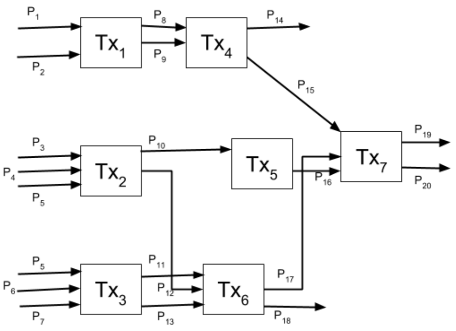

Fig. 8. Blockchain analysis - Transaction graph

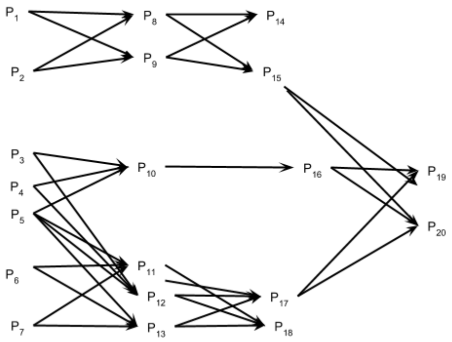

Fig. 9. Blockchain analysis - Address graph

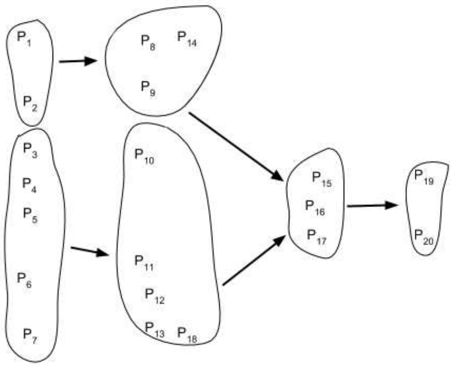

Fig. 10. Blockchain analysis - Entity/User graph

anonymity of the Bitcoin users and the corresponding state-of-the-art solutions that are proposed to enhance the same.

A. Blockchain Analysis and Deanonymization

A complete anonymity in Bitcoin is a complicated issue. To enforce anonymity in transactions, the Bitcoin system allows its users to generate multiple sets of public keys and it only stores the mapping information of a user to her public keys on the user's device. As a user can have multiple addresses, hence an adversary who is trying to deanonymize a user in a Bitcoin system needs to construct a one-to-many mapping between the user and its associated public keys. In particular, the Bitcoin users can be linked to a set of public addresses by using a detailed blockchain analysis procedure [110]. Authors in [65] show that the two non-trivial networking topologies called *transaction network* and *user network*, which provides reciprocal views of the Bitcoin system and have possible adverse implications for user anonymity. Similar to the work done in [65], authors in [111] presents an evaluation for privacy concerns in Bitcoin systems by analyzing the public blockchain. The analysis of blockchain requires three pre-processing steps, which include:

- *Transaction graph*: The whole blockchain could be viewed as an acyclic *transaction graph* $G_t = \{T, E\}$, where T is a set of transactions stored in the blockchain, and E is the set of unidirectional edges between these transactions. A G_t represents the flow of coins between transactions in the blockchain over time. The set of input and output coins in a transaction can be viewed as the weights on the edges in a G_t. In particular, each incoming edge $e \in E$ in a transaction carries a timestamp and the number of coins (C_i) that forms an input for that transaction. Figure 8 shows an instance of transaction graph in a blockchain.
- *Address graph*: By traversing the transaction graph we can easily infer the relationship between various input and output addresses (i.e., public keys), and using these relations we can generate an *address graph*, $G_a = \{P, E'\}$, where P is the set of Bitcoin addresses and E' are the edges connecting these addresses. Figure 9 shows an address graph derived from Figure 9.
- *User/entity graph*: By using the address graph along with a number of heuristics which are derived from Bitcoin protocol, the next step is to create an *entity graph* by grouping addresses that seem to belong to the same user. The entity graph, $G_e = \{U, E''\}$, where U is a disjoint subset of public keys (p) such that $p \in P$ and E'' are the edges connecting different $U's$ to show a directed connectivity between them. Figure 10 shows the entity graph derived from Figure 9 based on a set of heuristics.

In [111], authors introduce two heuristics that are derived directly from Bitcoin protocols or its common practices. The first is the most widely used heuristic that provides an adequate level of linkability and it heavily depends on the implementation details of Bitcoin protocols, and are termed as *idioms of use* as mentioned in [112]. The *idioms of use* assumes that all the inputs in a transaction are generated by the same user because in practice different users rarely contribute in a single, collaborative transaction. This heuristics also supports the fact that transitive closure can be applied to the transaction graph to yield clusters of Bitcoin addresses. For instance, by applying the above heuristic along with its transitive property on Figure 8, one can assume that transactions Tx_2 and Tx_3 are initiated by the same user as both shares a common input p_5, hence the addresses ranging from p_3 to p_6 could belong to one user. The second heuristic links the input addresses of a transaction to its output addresses by assuming that these outputs as *change* addresses if an output address is completely new (i.e., the address has never appeared in the past and it will not be seen in the blockchain to be re-used to receive payments). In Figure 9, the addresses p_{14} and p_{18} satisfy the second heuristic, and thus these addresses can be clustered with their inputs as shown in the Figure 10. Authors in [112] argued that the aforementioned heuristics are prone to errors, in cases where the implementation of Bitcoin protocols change with time, and the traditional Bitcoin network also changes which now consists of more number of mining pools instead of single users. Due to these facts, it is possible that the entity graph might contain a large number of false positives in the clustering process, hence it leads to the further refinements in the above heuristics. To reduce these false positives, authors in [112] suggest the manual inspection process identify the usage patterns induced by Bitcoin services (such as SatoshiDice). For instance, SatoshiDice requires that the payouts use the same address, therefore if a user spent coins using a change address, the address would receive another input which invalidates the one-time receive property of a change address. Furthermore, in [94] authors exploit the multi-signature addressing technique for the purpose of adverse effect to the user privacy. Authors conclude that even if the bitcoin addresses are changed, the structure of the *change* address in a multi-signature transaction can be matched to its input addresses.

Apart from using the adaptable and refined heuristics to match with the constantly changing blockchain usage patterns and Bitcoin services, the adversary needs to take further steps to link the address clusters with the real-world identities once an entity graph with low false positives is created. Authors in [112] perform with high precision the linking of clusters with the online wallets, vendors, and other service providers as one can do several interactions with these entities and learn at least one associated address. However, identifying regular users is difficult with the same approach, but the authors also suggest that authorities with subpoena power might even be able to identify individual users since most of the transaction flow passes through their centralized servers. These servers usually require keeping records for customer identities. Furthermore, the use of side-channel information is considered helpful in mapping the addresses. For instance, WikiLeaks, Silk Road, to name a few, uses publicly known addresses, and many service providers such as online sellers or exchange services require the user identity before providing a service. One can also make use of the web crawlers (such as bitcointalk.org) that searches the social networks for Bitcoin addresses [113] [114].

A commercial approach for blockchain analysis could be to use the software BitIodine [115] that offers an automated blockchain analysis framework. Due to its rapid growth in such a short span of time, the Bitcoin networks has become of great interest to governments and law enforcement agencies all over the world to track down the illicit transactions. By predicting that their is a huge market potential for Bitcoin, various companies such as Elliptic, Chainalysis, Numisight, Skry, to name a few, are specializing in "bitcoin blockchain analysis". These companies provide a set of tools to analyze the blockchain to identify illicit activities and even help to identify the Bitcoin users in the process. Authors in [116] propose *BitConeView*, a graphical tool for the visual analysis of coin flows in a blockchain. BitConeView allows to graphically track how Bitcoins from the given sources (i.e., transaction inputs) are spent over time by means of transactions and are eventually stored at multiple destinations (i.e., unspent transaction outputs).

Finally, network de-anonymization could be used to link an IP address to a user in the Bitcoin's peer-to-peer (P2P) network because while broadcasting a transaction the node leaks their IP address. Same as the blockchain analysis, a rigorous way to link IP addresses to hosts is by exploiting the network related information that can be collected by just observing the Bitcoin network. Over the years, multiple deanonymization attacks in which an adversary uses a "supernode" that connects with the active peers and listens to the transaction traffic relayed by honest nodes in the Bitcoin P2P network [66] [117] [65] are proposed. By exploiting the symmetric diffusion of transactions over the network, it is possible to link the Bitcoin users' public keys to their IP addresses with an accuracy of nearly 30% [66]. Moreover, the use of "supernode" for linking is trivial, hence exploits only minimal knowledge of the P2P graph structure and the structured randomness of diffusion. Therefore, we can hypothesize that even higher accuracies could be achieved by using the more sophisticated network traffic analyzing techniques.

B. Proposals for enabling privacy and improving anonymity

Privacy is not defined as an inherent property in Bitcoin's initial design, but it is strongly associated with the Bitcoin system. Therefore, in the recent years, an array of academic research [111] [133] [134] [115] which shows various privacy-related weaknesses in the current Bitcoin protocol has been surfaced. This research triggered a large set of privacy-enhancing technologies [133] [127] [59] [123] [125] [135] [118] [122] aiming at strengthening privacy and improving anonymity in the Bitcoin system without breaking its fundamental design principles. In this section, we discuss these state-of-the-art protocols which work toward the enhancement of privacy and anonymity in Bitcoin systems.

Based on the aforementioned discussion in Section V, it is evident that the public nature of the blockchain poses a significant threat to the privacy of Bitcoin users. Even worse, since funds can be tracked and tainted, no two coins are equal, and fungibility, a fundamental property required in every currency, is at risk. With these threats in mind, several privacy-enhancing technologies have been proposed to improve transaction privacy in Bitcoin. The state-of-the-art proposals (refer Table III) for enabling privacy in Bitcoin can be broadly classified into the following three categories:

- *Peer-to-peer mixing protocols.* In peer-to-peer (P2P) mixing protocols [136] [121] [69], a set of untrusted Bitcoin users simultaneously broadcast their messages to create a series of transactions without requiring any trusted third party. The main feature of a P2P mixing protocol is to ensure sender anonymity within the set of participates by permuting ownership of their coins. The goal is to prevent an attacker which controls a part of the network or some of the participating users to associate a transaction to its corresponding honest sender. The degree of anonymity in P2P protocols depend on the number of users in the anonymity set. Table III shows a range of P2P mixing protocols along with their brief description, advantages, and disadvantages in terms of user anonymity and transaction security. CoinJoin [68], a straightforward protocol for implementing P2P mixing which aims to enhance privacy and securely prevent thefts. In CoinJoin, a set of users with agreed (via their primary signatures) inputs and outputs create a standard Bitcoin transaction such that no external adversary knows which output links with which input, hence it ensures external unlinkability. To prevent theft, a user only signs the transaction if its desired output appears in the output addresses of the transaction. In this

TABLE III
TECHNIQUES FOR IMPROVING PRIVACY AND ANONYMITY IN BITCOIN

Proposals	**Type/Class**	**Distinct features and properties**	**Advantages**	**Disadvantages**
CoinJoin [68]	P2P	uses multi-signature transactions to enhance privacy	prevent thefts, lower per-transaction fee	anonymity level depends on the number of participants, vulnerable to DoS, sybil and intersection attacks, prevents plausible deniability
CoinShuffle [69]	P2P	decentralized protocol for coordinating CoinJoin transactions through a cryptographic mixing protocol	internal unlinkability, robust to DoS attacks, theft resistance	lower anonymity level and deniability, prone to intersection and sybil attacks
Xim [59]	P2P	anonymously partnering and multi-round mixing	distributed pairing, internal unlinkability, thwarts sybil and DoS attacks	higher mixing time
CoinShuffle++ / DiceMix [118]	P2P	based on CoinJoin concept, optimal P2P mixing solution to improve anonymity in crypto-currencies	low mixing time (8 secs for 50 peers), resistant to deanonymization attack, ensures sender anonymity and termination	vulnerable to DoS and sybil attacks, limited scalability, no support for Confidential Transactions (CT)
ValueShuffle [119]	P2P	based on CoinShuffle++ concept, uses Confidential Transactions mixing approach to achieve comprehensive transaction privacy	unlinkability, CT compatibility and theft resistance, normal payment using ValueShuffle needs only one transaction	vulnerable to DoS and sybil attacks, limited scalability
Dandelion [120]	P2P	networking policy to prevent network-facilitated deanonymization of Bitcoin users	provides strong anonymity even in the presence of multiple adversaries	vulnerable to DoS and sybil attacks
SecureCoin [121]	P2P	based on CoinParty concept, an efficient and secure protocol for anonymous and unlinkable Bitcoin transactions	protect against sabotage attacks, attempted by any number of participating saboteurs, low mixing fee, deniability	vulnerable to DoS attacks, limited scalability
CoinParty [122]	partially P2P	based on CoinJoin concept, uses threshold ECDSA and decryption mixnets to combine pros of centralized and decentralized mixes in a single system	improves on robustness, anonymity, scalability and deniability, no mixing fee	partially prone to coin theft and DoS attack, high mixing time, requires separate honest mixing peers
MixCoin [123]	Distributed	third-party mixing with accountability	DoS and sybil resistance	partial internal unlinkability and theft resistance,
BlindCoin [124]	Distributed	based on MixCoin concept, uses blind signature scheme to ensure anonymity	internal unlinkability, DoS and sybil resistance	partial theft resistance, additional costs and delays in mixing process
TumbleBit [125]	Distributed	undirectional unlinkable payment hub that uses an untrusted intermediary	prevents theft, anonymous, resists intersection, sybil and DoS, scalable (implemented with 800 users)	normal payment using TumbleBit needs at least two sequential transactions
ZeroCoin / ZeroCash [126] [127]	Altcoin	a cryptographic extension to Bitcoin , unlinkable and untraceable transactions by using zero knowledge proofs	provides internal unlinkability, theft and DoS resistance	relies on a trusted setup and non-falsifiable cryptographic assumptions, blockchain pruning is not possible
CryptoNote [128]	Altcoin	relies on ring signatures to provide anonymity	provides strong privacy and anonymity guarantees	higher computational complexity, not compatible with pruning
MimbleWimble [129] [130]	Altcoin	a design for a cryptocurrency with confidential transactions	CT compatibility, improve privacy, fungibility and scalability	vulnerable to DoS attacks, not compatible with smart contracts
ByzCoin [131]	Altcoin	Bitcoin-like cryptocurrency with strong consistency via collective signing	lower consensus latency and high transaction throughput, resistance to selfish and stubborn mining [132], eclipse and delivery-tampering and double-spending attacks	vulnerable to slow down or temporary DoS attack and 51% attack,

way, CoinJoin makes the multiple inputs of a transaction independent from each other, thus it breaks the basic heuristic from Section V-A (i.e., inputs of a transaction belong to the same Bitcoin user). However, CoinJoin has few major drawbacks which include, limited scalability and privacy leakage due to the need of managing signatures of the involved participants in the mixing set, the requirement of signing a transaction by all its participants make CoinJoin vulnerable to DoS attacks, and to create a mix each participant has to share their signature and output addresses within the participating set which causes internal unlinkability. To address the aforementioned internal unlinkability issue and to increase the robustness to DoS attacks, authors in [69] propose CoinShuffle, a decentralized protocol that coordinates CoinJoin transactions using a cryptographic mixing technique. Later, an array of protocols [118] [119] [121] are built on the concept of either CoinJoin or CoinShuffle that enhances the P2P mixing by providing various improvements that include, resistance to DoS, sybil, and intersection attacks, plausible deniability, low mixing time, and scalability of the mixing groups.

- *Distributed mixing networks.* Authors in [123] propose *MixCoin*, a third-party mixing protocol to facilitate anonymous payments in Bitcoin and similar cryptocurrencies. The *MixCoin* uses the emergent phenomenon of currency mixes, in which a user shares a number of coins with a third-party mix using a standard-sized transaction, and it receives back the same number of coins from the mix that is submitted by some other user, hence it provides strong anonymity from external entries. *MixCoin* uses a reputation-based cryptographic accountability technique to prevent other users within the mix from theft and disrupting the protocol. However, mixes might steal the user coins at any time or become a threat to the user anonymity because the mix will know the internal mapping between the users and outputs. To provide internal unlinkability (i.e., preventing the mix from learning input-output linking) in *MixCoin*, authors in [124] proposes *BlindCoin* which extends the *MixCoin* protocol by using blind signatures to create user inputs and cryptographically blinded outputs called *blinded tokens*. However, to achieve this internal unlinkability, *BlindCoin* requires two extra transactions to publish and redeem the blinded tokens, and the threat of theft from mix is still present. Recently, in [125] authors propose *TumbleBit*, a Bitcoin-compatible unidirectional unlinkable payment hub that allows peers to make fast, off-blockchain payments anonymously over an untrusted intermediary called *Tumbler*. Similar to Chaumian original eCash protocol [137], TumbleBit enforces anonymity in the mixing by ensuring that no one, not even the Tumbler can link a transaction to its sender to its receiver. The mixing of payments from 800 users show that TumbleBit provides strong anonymity and theft resistance, and it is scalable.
- *Bitcoin extensions or Altcoins.* Instead of proposing techniques (such as mixing and shuffling) to increase transaction anonymity and user privacy in Bitcoin, there are also mechanisms which work as, an extension to Bitcoin or a full-fledged altcoin. Authors in [126] propose *ZeroCoin*, a cryptographic extension to Bitcoin which provides anonymity by design by applying zero knowledge proofs (ZKP). In ZeroCoin, a user can simply wash the linkability traces from its coins by exchanging them for an equal value of ZerCoins. But unlike the aforementioned mixing approaches, the user should not have to ask for the exchange to a mixing set, instead, the user can itself generate the ZeroCoins by proving that she owns the equal value of Bitcoins via the Zerocoin protocol. Zerocoin currently derives both its anonymity and security against counterfeiting from strong cryptographic assumptions at the cost of substantially increased computational complexity and size. The use of zero-knowledge proofs prevent the transaction graph analyses. An extension of ZeroCoin called *ZeroCash* is presented by [127]. ZeroCash uses an improved version of ZKP (in terms of functionality and efficiency) called SNARKs, which hides additional information about transactions such as the amount and recipient addresses to achieve strong privacy guarantees. However, ZeroCash relies on a trusted setup for generation of secret parameters required for SNARKs implementation, it requires protocol modifications, and the blockchain pruning is not possible. Recently, authors in [129] propose *MimbleWimble*, an altcoin that supports confidential transactions (CT). The CTs can be aggregated non-interactively and even across blocks, thus greatly increases the scalability of the underlying blockchain. However, such aggregation alone does not ensure input-output unlinkability against parties who perform the aggregation, e.g., the miners. Additionally, Mimblewimble is not compatible with smart contracts due to the lack of script support.

As a summary, in this section, the Bitcoin system's privacy and anonymity concerns are discussed. It is observed that Bitcoin is pseudo-anonymous, as the account is tied to the keys and not to the individual users. As the need of Bitcoins increases, the need for privacy and anonymity protection also increases, and it must be ensured that the users will receive a satisfactory level of service in terms of privacy, security, and anonymity.

VI. Research directions in security and privacy of Bitcoins

In this section, we discuss various issues and open challenges to formulate possible future research directions in Bitcoin. Some of the directives are already discussed in the previous sections. However, remaining challenges are dealt in brief in this section.

- *Game theory and stability:* Recall that mining pools consist of individual miners who pool their hashing power as well as their incentives. Miners can behave selfishly by holding on to their blocks and releasing it whenever they want. This kind of selfish behavior may pose a game theoretic problem between the selfish miners and the network. Since all the miners perform with a notion of increasing their incentives, a game theoretic approach is well suited for achieving Nash equilibrium among miners (i.e., players) [138]. Attackers may try to contribute to an increase of their chain length compared to honest chain in the network. This poses a game between the honest chain miners and the malicious miners, thus achieving equilibrium to bring stability in the network is a possible research direction. There are numerous proposals [138] [139] [140] which shows that the use of the game-theoretic approaches provide useful information about the effects of selfish mining, block withholding and discarding attacks, and the incentive distribution problem in the mining pools. Therefore, we believe that this approach could be effectively used for modeling the various issues and providing adequate solutions for the identified issues related to the mining pools.
- *Cryptographic and keying techniques:* The Simplified Payment Verification (SPV) protocol which is a light weight protocol used for the verification of the transaction

sent from a user [141], and it is often vulnerable to attacks like sybil and double spending. A more robust verification protocol is a current requirement. For the key manipulations and calculations, a distributed approach is always preferred more than the centralized one. This is to avoid the point of failure or the central server under the risk of an attack. Hence, in this direction, the innovative means of key computation and storage of the Bitcoins in a distributed fashion is a possible research direction. Additionally, the Bitcoin protocols use EDCSA and hash functions like SHA-256 which creates another research scope as there is always an adequate requirement to improve these algorithms or implement novel keying and hashing techniques. We have seen the use of cluster or group keys which are based on some threshold in order to solve various attacks. For instance, fix a group head and get an additional signature or authentication on every transaction [133]. Another approach is to use "trusted paths" which is based on hardware that allows users to read and write a a few cryptographic data [133]. Finally, there are few techniques which use Bloom filters for securing wallets. Nevertheless, filters might lead to false positives and false negatives that will consume the network bandwidth, thus reducing it can be a potential research directive.

- *Improving blockchain protocol:* Blockchain provides for the first time a probabilistic solution to the Byzantine Generals problem [142], where consensus is reached over time (after confirmations) and makes use of economic incentives to secure the functionality of the overall infrastructure. The blockchain technology promises to revolutionize the way we conduct business. For instance, blockchain startups have received more than one billion dollars [143] of venture capital money to exploit this technology for applications such as voting, record keeping, contracts, to name a few. Despite its potential, blockchain protocol faces significant concerns in terms of its privacy [144] and scalability [103] [145]. The append-only nature of the blockchain is essential to the security of the Bitcoin ecosystem as transactions are stored in the ledger forever and are immutable. However, an immutable ledger is not appropriate for all new applications that are being envisaged for the blockchain. Recently, authors in [146] present modification in blockchain techniques that allows operation such as re-writing one or more blocks, compressing any number of blocks into a smaller number of blocks, and inserting one or more blocks.
- *Fastness:*Bitcoin's proof of work is designed to validate a new block on average every 10 minutes, and it is recommended to wait for six confirmations before accepting a transaction [147], which makes it impractical for many real-world applications (e.g., a point of sale payments). Faster mining with the same robustness such as one proposed in [131] is a future requirement. Recently authors in [148] present *Proof of Luck*, an efficient blockchain consensus protocol to achieve low-latency transaction validation, deterministic confirmation time, negligible energy consumption, and equitably distributed mining.
- *Incentives for miners:* In general, incentives can be either fixed or variable depending on the complexity of the puzzle that miners solve. A variable incentive may increase the competition between the miners and help to solve puzzles that are challenging. The miners who inform the malfunctions and other illegal behavior in the network can be awarded additional coins as a reward. This act will increase the number of honest nodes in the network. In the world of growing demand for the cryptocurrencies, there is a lot of competition for Bitcoins or any other digital currency to retain its popularity in the market. Additionally, miners may migrate by looking at the rewards given by the other competitors or by the fact that for every four years the incentives are halved. Therefore, essential questions that need addressing includes, how to make the miners fix to a currency in such a competitive environment, and what are the other incentives the Bitcoin system can think of to attract the miners.
- *Smart contracts and preventing backtracks:* Smart contract refers to the computer programs that embody a self-executing and self-enforcing contract to which users may become a party, by interacting with it electronically. These contracts are of particular interest to those in the financial sector. However, the concept of smart contract is not a new one, but the advent of blockchain technology spurred interest in it because the blockchain eliminates the need to rely on a trusted third party to "execute" the contract, and enables to use of cryptocurrency as "programmable money". Bitcoins support for smart contracts is extremely limited. Recently authors in [149] propose *Hawk*, which uses a blockchain model of cryptography to generate privacy-preserving smart contracts. Similar to Bitcoins, authors in [150] proposes *Enigma*, a decentralized computation platform which provides a highly optimized version of secure multi-party computation with guaranteed privacy to effectively execute smart contracts.

VII. CONCLUSIONS

Bitcoins have already evinced as a popular digital currency in the market. However, the fame of Bitcoin has attracted antagonists to use Bitcoin network for their selfish motives and benefits. Today we have nearly 700 different cryptocurrencies in action, nevertheless, the outstanding popularity of Bitcoin makes this currency favorite for hackers. According to our survey, even though the construction of the Bitcoin protocols with proof-of-work and consensus to protect the user actions are the robust features of Bitcoin, these itself becoming a point of manipulation for cyber thieves. Starting from packet sniffing to the double spending, the Bitcoin systems are dreaded with various attacks. Though literature provides solutions against few of these attacks, but the robust and effective security solutions that can ensure proper functioning of the Bitcoin in the future are still absent. Together with security, the distributed nature of Bitcoin's blockchain protocols has lead glitches in the privacy and anonymity requirements of the users. In summary, this paper is a sole attempt towards highlighting the security and privacy issues in different fields of Bitcoin. Once presenting the major components of Bitcoin, its basic characteristics and related concepts, in brief, our survey mainly focuses on the security and privacy aspects that can be found at various stages in the Bitcoin system, starting from transaction creation to its successful addition in the blockchain. We studied and emphasize the issue of user privacy and anonymity in this rapidly growing e-commerce industry. With the set of future research directions and open questions that we have raised, we hope that our work will motivate fledgling researchers towards tackling the security and privacy issues of Bitcoin systems.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Available: http://bitcoin.org/bitcoin.pdf*, 2008.

[2] 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*, May 2015, pp. 104–121.

[3] WikiLeaks, "Donation request via cyrptocurriencies," *Available: https://shop.wikileaks.org/donate*.

[4] W. F. Slater, "Bitcoin: A current look at the worlds most popular, enigmatic and controversial digital cryptocurrency," in *A Presentation for Forensecure 2014*, April 2014.

[5] "Status about bitcoin technoogy was obtained fromwhat 2016 holds for bitcoin business," *Available: http://www.coindesk.com/what-2016-holds-for-bitcoin-businesses/*.

[6] M. T. Alam, H. Li, and A. Patidar, "Bitcoin for smart trading in smart grid," in *The 21st IEEE International Workshop on Local and Metropolitan Area Networks*, April 2015, pp. 1–2.

[7] Y. Zhang and J. Wen, "An iot electric business model based on the protocol of bitcoin," in *2015 18th International Conference on Intelligence in Next Generation Networks*, Feb 2015, pp. 184–191.

[8] A. Bahga and V. K. Madisetti, "Blockchain platform for industrial internet of things," *Available: http://file.scirp.org/pdf/JSEA_2016102814012798.pdf*, 2016.

[9] S. Rowan, M. Clear, M. Gerla, M. Huggard, and C. Mc Goldrick, "Securing vehicle to vehicle communications using blockchain through visible light and acoustic side-channels," *Available: https://arxiv.org/abs/1704.02553*, 2017.

[10] C. Adhikari, "Secure framework for healthcare data management using ethereum-based blockchain technology," *Available: http://scholarworks.boisestate.edu/eng_17/14/*, 2017.

[11] K. Biswas and V. Muthukkumarasamy, "Securing smart cities using blockchain technology," in *2016 IEEE 18th International Conference on High Performance Computing and Communications (HPCC/SmartCity/DSS)*, Dec 2016, pp. 1392–1393.

[12] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2016.

[13] S. your wallet:, "The bitcoin wiki," *Available: https://en.bitcoin.it/wiki/Securing_your_wallet*, Mar. 2014.

[14] G. Andresen, "Bip 16: Pay to script hash," *Available: https://github.com/bitcoin/bips/blob/master/bip-0016.mediawik*, Jan. 2012.

[15] J. R. Douceur, "The sybil attack," in *the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS '01. London, UK: Springer-Verlag, 2002, pp. 251–260.

[16] N. T. Courtois, M. Grajek, and R. Naik, "The unreasonable fundamental incertitudes behind bitcoin mining," *CoRR*, vol. abs/1310.7935, 2013.

[17] D. E. III and T. Hansen, "Us secure hash algorithms (sha and sha-based hmac and hkdf)," *Available: http://www.ietf.org/rfc/rfc6234.txt*, 2011.

[18] K. Kaskaloglu, "Near zero bitcoin transaction fees cannot last forever," *Proc. Int. Conf. Digit. Secur. Forensics*, pp. 91–99, 2014.

[19] D. Kraft, "Difficulty control for blockchain-based consensus systems," *Peer-to-Peer Networking and Applications*, vol. 9, no. 2, pp. 397–413, 2016.

[20] R. C. Merkle, *A Digital Signature Based on a Conventional Encryption Function*. Springer Berlin Heidelberg, 1988, pp. 369–378.

[21] M. Rosenfeld, "Analysis of bitcoin pooled mining reward systems," *CoRR*, vol. abs/1112.4980, 2011.

[22] "Comparison of existing bitcoin pools and their reward strategies," *Available: https://en.bitcoin.it/wiki/Comparisonofminingpools*.

[23] "Discussion of different pooled mining approaches," *Available: https://en.bitcoin.it/wiki/Pooledmining*.

[24] M. Kiran and M. Stannett, "Bitcoin risk analysis," *Available: http://www.nemode.ac.uk/wp-content/uploads/2015/02/2015-Bit-Coin-risk-analysis.p* Dec. 2014.

[25] B. Masooda, S. Beth, and B. Jeremiah, "What motivates people to use bitcoin?" in *Social Informatics: 8th International Conference, SocInfo 2016*. Springer International Publishing, 2016, pp. 347–367.

[26] K. Krombholz, A. Judmayer, M. Gusenbauer, and E. Weippl, "The other side of the coin: User experiences with bitcoin security and privacy?" *Available: https://www.sba-research.org/wp-content/uploads/publications/TheOtherSideOfTheC* 2016.

[27] G. O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in bitcoin," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, ser. CCS '12. New York, NY, USA: ACM, 2012, pp. 906–917.

[28] G. O. Karame, E. Androulaki, M. Roeschlin, A. Gervais, and S. Čapkun, "Misbehavior in bitcoin: A study of double-spending and accountability," *ACM Trans. Inf. Syst. Secur.*, vol. 18, no. 1, May 2015.

[29] T. Bamert, C. Decker, L. Elsen, R. Wattenhofer, and S. Welten, "Have a snack, pay with bitcoins," in *IEEE P2P 2013 Proceedings*, Sept 2013, pp. 1–5.

[30] L. Bahack, "Theoretical bitcoin attacks with less than half of the computational power (draft)," *CoRR*, vol. abs/1312.7013, 2013.

[31] H. Finney, "Best practice for fast transaction acceptancehow high is the risk?" *Available: https://bitcointalk.org/index.php?topic=3441.msg48384#msg48384*, 2011.

[32] J. Heusser, "Sat solvingan alternative to brute force bitcoin mining," *Available: https://jheusser.github.io/2013/02/03/satcoin.html*, 2013.

[33] Vector67, "Fake bitcoins?" *Available: https://bitcointalk.org/index.php?topic=36788.msg463391#msg463391*, 2011.

[34] J. A. Kroll, I. C. Davey, and E. W. Felten, "The economics of bitcoin mining, or bitcoin in the presence of adversaries," 2013.

[35] I. Eyal and E. G. Sirer, "How to disincentivize large bitcoin mining pools," *Available: http://hackingdistributed.com/2014/06/18/how-to-disincentivize-large-bitcoin-mining* 2014.

[36] M. Bastiaan, "Preventing the 51%-attack: a stochastic analysis of two phase proof of work in bitcoin," Jan 2015.

[37] A. Chepurnoy, T. Duong, L. Fan, and H.-S. Zhou, "Twinscoin: A cryptocurrency via proof-of-work and proof-of-stake," 2017, http://eprint.iacr.org/2017/232.

[38] P. Daian, I. Eyal, A. Juels, and G. Sirer, "Piecework: Generalized outsourcing control for proofs of work," In BITCOIN Workshop, 2017.

[39] N. T. Courtois and L. Bahack, "On subversive miner strategies and block withholding attack in bitcoin digital currency," *CoRR*, vol. abs/1402.1718, 2014.

[40] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," *CoRR*, vol. abs/1311.0243, 2013.

[41] R. Zhang and B. Preneel, *Publish or Perish: A Backward-Compatible Defense Against Selfish Mining in Bitcoin*. Springer International Publishing, 2017, pp. 277–292.

[42] S. Solat and M. Potop-Butucaru, "Zeroblock: Preventing selfish mining in bitcoin," *CoRR*, vol. abs/1605.02435, 2016.

[43] E. Heilman, *One Weird Trick to Stop Selfish Miners: Fresh Bitcoins, A Solution for the Honest Miner*. Springer Berlin Heidelberg, 2014.

[44] S. D. Lerner, "Decor+," *Available: https://bitslog.wordpress.com/2014/05/07/decor-2/*, 2014.

[45] S. Bag, S. Ruj, and K. Sakurai, "Bitcoin block withholding attack : Analysis and mitigation," *IEEE Transactions on Information Forensics and Security*, vol. PP, no. 99, pp. 1–12, 2016.

[46] B. J., "Why buy when you can rent?" *Financial Cryptography and Data Security. FC 2016. Lecture Notes in Computer Science, vol 9604. Springer, Berlin, Heidelberg*, 2016.

[47] P. McCorry, S. F. Shahandashti, and F. Hao, "Refund attacks on bitcoins payment protocol," 2016, http://eprint.iacr.org/2016/024.

[48] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton, NJ, USA: Princeton University Press, 2016.

[49] A. Miller, "Feather-forks: enforcing a blacklist with sub-50% hash power," *Available: https://bitcointalk.org/index.php?topic=312668.0*, 2013.

[50] M. Andrychowicz, S. Dziembowski, D. Malinowski, and Ł. Mazurek, *On the Malleability of Bitcoin Transactions*. Springer Berlin Heidelberg, 2015, pp. 1–18.

[51] C. Decker and R. Wattenhofer, *Bitcoin Transaction Malleability and MtGox*. Springer International Publishing, 2014, pp. 313–326.

[52] P. Wuille, "Bip 62: Dealing with malleability," *Available: https://github.com/bitcoin/bips/blob/master/bip-0062.mediawiki*, Mar. 2014.

[53] R. Gennaro, S. Goldfeder, and A. Narayanan, *Threshold-Optimal DSA/ECDSA Signatures and an Application to Bitcoin Wallet Security*. Springer International Publishing, 2016, pp. 156–174.

[54] S. Goldfeder, J. Bonneau, E. W. Felten, J. A. Kroll, and A. Narayanan, "Securing bitcoin wallets via threshold signatures," *Available: http://www.cs.princeton.edu/stevenag/bitcoin_threshold_signatures.pdf*, 2014.

[55] T. Bamert, C. Decker, R. Wattenhofer, and S. Welten, *BlueWallet: The Secure Bitcoin Wallet*. Springer International Publishing, 2014, pp. 65–80.

[56] M. Gentilal, P. Martins, and L. Sousa, "Trustzone-backed bitcoin wallet," in *Proceedings of the Fourth Workshop on Cryptography and Security in Computing Systems*, ser. CS2 '17. New York, NY, USA: ACM, 2017, pp. 25–28.

[57] corbixgwelt, "Timejacking and bitcoin," *Available: http://culubas.blogspot.de/2011/05/timejacking-bitcoin_802.html*, Mar. 2011.

[58] D. Mills, J. Martin, J. Burbank, and W. Kasch, "Network time protocol version 4: Protocol and algorithms specification, rfc 5905, internet engineering task force," *Available: http://www.ietf.org/rfc/rfc5905.txt*, Mar. 2011.

[59] G. Bissias, A. P. Ozisik, B. N. Levine, and M. Liberatore, "Sybil-resistant mixing for bitcoin," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, ser. WPES '14. ACM, 2014, pp. 149–158.

[60] M. Vasek, M. Thornton, and T. Moore, *Empirical Analysis of Denial-of-Service Attacks in the Bitcoin Ecosystem*. Springer Berlin Heidelberg, 2014, pp. 57–71.

[61] B. Johnson, A. Laszka, J. Grossklags, M. Vasek, and T. Moore, *Game-Theoretic Analysis of DDoS Attacks Against Bitcoin Mining Pools*. Springer Berlin Heidelberg, 2014, pp. 72–86.

[62] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract]y," *SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 3, pp. 34–37, Dec. 2014.

[63] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in *Proceedings of the 24th USENIX Conference on Security Symposium*, ser. SEC'15. USENIX Association, 2015, pp. 129–144.

[64] A. Gervais, H. Ritzdorf, G. O. Karame, and S. Capkun, "Tampering with the delivery of blocks and transactions in bitcoin," in *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15. ACM, 2015, pp. 692–705.

[65] P. Koshy, D. Koshy, and P. McDaniel, *An Analysis of Anonymity in Bitcoin Using P2P Network Traffic*. Springer Berlin Heidelberg, 2014, pp. 469–485.

[66] A. Biryukov, D. Khovratovich, and I. Pustogarov, "Deanonymisation of clients in bitcoin p2p network," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '14. ACM, 2014, pp. 15–29.

[67] G. Danezis, R. Dingledine, and N. Mathewson, "Mixminion: design of a type iii anonymous remailer protocol," in *2003 Symposium on Security and Privacy, 2003.*, May 2003, pp. 2–15.

[68] G. Maxwell, "Coinjoin: Bitcoin privacy for the real world," *Available: https://bitcointalk.org/index.php?topic=279249.0*, Mar. 2013.

[69] T. Ruffing, P. Moreno-Sanchez, and A. Kate, *CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin*. Springer International Publishing, 2014, pp. 345–364.

[70] S. Haber and W. S. Stornetta, "How to time-stamp a digital document," *Journal of Cryptology*, vol. 3, no. 2, pp. 99–111, 1991.

[71] D. Malkhi, "Byzantine quorum systems," *Distrib. Comput.,*, vol. 4, p. 203213, Jan. 2012.

[72] N. Szabo, "Secure property titles with owner authority," *Available: http://nakamotoinstitute.org/secure-property-titles/*, 1988.

[73] C. Natoli and V. Gramoli, "The balance attack against proof-of-work blockchains: The R3 testbed as an example," *CoRR*, vol. abs/1612.09426, 2016.

[74] G. Wood, "Ethereum: A secure decentralised generalised transaction-ledger," *yellow paper*, 2015.

[75] M. Rosenfeld, "Mining pools reward methods," *Presentation at Bitcoin 2013 Conference*, 2013.

[76] L. Luu, R. Saha, I. Parameshwaran, P. Saxena, and A. Hobor, "On power splitting games in distributed computation: The case of bitcoin pooled mining," in *2015 IEEE 28th Computer Security Foundations Symposium*, July 2015, pp. 397–411.

[77] A. F. Neil Gandal, Tyler Moore and J. Hamrick, "The impact of ddos and other security shocks on bitcoin currency exchanges: Evidence from mt. gox," *The 15th Annual Workshop on the Economics of Information Security*, vol. abs/1411.7099, June 13-14, 2016.

[78] V. S. Miller, "Use of elliptic curves in cryptography," in *Lecture Notes in Computer Sciences; 218 on Advances in cryptology—CRYPTO 85*. Springer-Verlag New York, Inc., 1986, pp. 417–426.

[79] P. Gallagher and C. Kerry, "Federal information processing standards (fips) publication 186-4: Digital signature standard (dss)," *Available: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf*, July, 2013.

[80] N. A. Howgrave-Graham and N. P. Smart, "Lattice attacks on digital signature schemes," *Designs, Codes and Cryptography*, vol. 23, no. 3, pp. 283–290, 2001.

[81] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow, *Elliptic Curve Cryptography in Practice*. Springer Berlin Heidelberg, 2014, pp. 157–175.

[82] I. Giechaskiel, C. Cremers, and K. B. Rasmussen, *On Bitcoin Security in the Presence of Broken Cryptographic Primitives*. Springer International Publishing, 2016, pp. 201–222.

[83] J. J. Hoch and A. Shamir, *On the Strength of the Concatenated Hash Combiner When All the Hash Functions Are Weak*. Springer Berlin Heidelberg, 2008, pp. 616–630.

[84] S. Eskandari, D. Barrera, E. Stobert, and J. Clark, "A First Look at the Usability of Bitcoin Key Management," 2015. [Online]. Available: http://people.inf.ethz.ch/barrerad/files/usec15-eskandari.pdf

[85] P. Litke and J. Stewart, "Cryptocurrency-stealing malware landscape," Technical report, Dell SecureWorks Counter Threat Unit, 2014.

[86] T. Moore and N. Christin, *Beware the Middleman: Empirical Analysis of Bitcoin-Exchange Risk*. Springer Berlin Heidelberg, 2013, pp. 25–33.

[87] G. G. Dagher, B. Bünz, J. Bonneau, J. Clark, and D. Boneh, "Provisions: Privacy-preserving proofs of solvency for bitcoin exchanges," in *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15. ACM, 2015, pp. 720–731.

[88] T. Neudecker, P. Andelfinger, and H. Hartenstein, "A simulation model for analysis of attacks on the bitcoin peer-to-peer network," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, May 2015, pp. 1327–1332.

[89] "Malleability attack a nuisance but bitcoin not broken, pundits say," *Available: http://www.financemagnates.com/cryptocurrency/news/malleability-attack-a-nuisance*

[90] "The bitcoin malleability attack how can it undermine the blockchains credibility?" *Available: http://www.coinwrite.org/*, 2017.

[91] G. O. Karame, E. Androulaki, and S. Capkun, "Two bitcoins at the price of one? double-spending attacks on fast payments in bitcoin," 2012, http://eprint.iacr.org/2012/248.

[92] "Solution to sybil attacks and 51*Available: https://letstalkbitcoin.com/blog/post/solution-to-sybil-attacks-and-51-attacks-in-dece*
2014.

[93] M. O. Rabin, "Transaction protection by beacons," *Journal of Computer and System Sciences*, vol. 27, no. 2, pp. 256 – 267, 1983.

[94] S. Goldfeder, R. Gennaro, H. Kalodner, J. Bonneau, J. A. Kroll, E. W. Felten, and A. Narayanan, "Securing bitcoin wallets via a new dsa-ecdsa threshold signature scheme," *Available: https://www.cs.princeton.edu/steve-nag/thresholdsigs.pdf.*, 2016.

[95] M. Draupnir, "Bitcoin cold storage guide," *Available: https://www.weusecoins.com/bitcoin-cold-storage-guide/*, Mar. 2016.

[96] "Biometric tech secures bitcoin wallet," *Biometric Technology Today*, vol. 2015, no. 6, 2015.

[97] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[98] I. Eyal, "The miner's dilemma," *CoRR*, vol. abs/1411.7099, 2014.

[99] P. Camelo, J. Moura, and L. Krippahl, "CONDENSER: A graph-based approachfor detecting botnets," *CoRR*, vol. abs/1410.8747, 2014.

[100] Mate, "How to identify transaction malleability attacks," *Available: https://news.bitcoin.com/identify-transaction-malleability-attacks/*, 2015.

[101] C. Decker and R. Wattenhofer, *A Fast and Scalable Payment Network with Bitcoin Duplex Micropayment Channels*. Springer International Publishing, 2015.

[102] B. Rosenberg, *Micropayment Systems. Handbook of Financial Cryptography and Security*, 1st ed. Chapman & Hall/CRC, 2010.

[103] Y. Sompolinsky and A. Zohar, "Accelerating bitcoin's transaction processing. fast money grows on trees, not chains," 2013, http://eprint.iacr.org/2013/881.

[104] J. A. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol with chains of variable difficulty," 2016, http://eprint.iacr.org/2016/1048.

[105] X. Min, Q. Li, L. Liu, and L. Cui, "A permissioned blockchain framework for supporting instant transaction and dynamic block size," in *2016 IEEE Trustcom/BigDataSE/ISPA*, Aug 2016, pp. 90–96.

[106] QuantumMechanic, "Proof of stake instead of proof of work," *Available: bitcointalk.org*, July 2011.

[107] J. Clark and A. Essex, *CommitCoin: Carbon Dating Commitments with Bitcoin*. Springer Berlin Heidelberg, 2012, pp. 390–398.

[108] IStewart, "Proof of burn," December 2012.

[109] S. Park, K. Pietrzak, A. Kwon, J. Alwen, G. Fuchsbauer, and P. Gai, "Spacemint: A cryptocurrency based on proofs of space," 2015, http://eprint.iacr.org/2015/528.

[110] D. Ron and A. Shamir, *Quantitative Analysis of the Full Bitcoin Transaction Graph*. Springer Berlin Heidelberg, 2013, pp. 6–24.

[111] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun, *Evaluating User Privacy in Bitcoin*. Springer Berlin Heidelberg, 2013, pp. 34–51.

[112] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A fistful of bitcoins: Characterizing payments among men with no names," in *Proceedings of the 2013 Conference on Internet Measurement Conference*, ser. IMC '13. ACM, 2013, pp. 127–140.

[113] M. Fleder, M. S. Kester, and S. Pillai, "Bitcoin transaction graph analysis," *CoRR*, vol. abs/1502.01657, 2015.

[114] F. Reid and M. Harrigan, *An Analysis of Anonymity in the Bitcoin System*. Springer New York, 2013, pp. 197–223.

[115] M. Spagnuolo, F. Maggi, and S. Zanero, *BitIodine: Extracting Intelligence from the Bitcoin Network*. Springer Berlin Heidelberg, 2014, pp. 457–468.

[116] 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 *IEEE Symposium on Visualization for Cyber Security (VizSec)*, Oct 2015, pp. 1–8.

[117] A. Biryukov and I. Pustogarov, "Bitcoin over tor isn't a good idea," in *2015 IEEE Symposium on Security and Privacy*, May 2015, pp. 122–134.

[118] T. Ruffing, P. Moreno-Sanchez, and A. Kate, "P2p mixing and unlinkable bitcoin transactions," NDSS'17, http://eprint.iacr.org/2016/824.

[119] T. Ruffing and P. Moreno-Sanchez, "Mixing confidential transactions: Comprehensive transaction privacy for bitcoin," 2017, http://eprint.iacr.org/2017/238.

[120] S. B. Venkatakrishnan, G. C. Fanti, and P. Viswanath, "Dandelion: Redesigning the bitcoin network for anonymity," *CoRR*, vol. abs/1701.04439, 2017.

[121] M. H. Ibrahim, "Securecoin: A robust secure and efficient protocol for anonymous bitcoin ecosystem," *I. J. Network Security*, vol. 19, pp. 295–312, 2017.

[122] J. H. Ziegeldorf, F. Grossmann, M. Henze, N. Inden, and K. Wehrle, "Coinparty: Secure multi-party mixing of bitcoins," in *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '15. ACM, 2015, pp. 75–86.

[123] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, *Mixcoin: Anonymity for Bitcoin with Accountable Mixes*. Springer Berlin Heidelberg, 2014, pp. 486–504.

[124] L. Valenta and B. Rowan, "Blindcoin: Blinded, accountable mixes for bitcoin," in *Financial Cryptography Workshops*, 2015.

[125] E. Heilman, L. Alshenibr, F. Baldimtsi, A. Scafuro, and S. Goldberg, "Tumblebit: An untrusted bitcoin-compatible anonymous payment hub," 2016, http://eprint.iacr.org/2016/575.

[126] 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*, May 2013, pp. 397–411.

[127] 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*, May 2014, pp. 459–474.

[128] N. van Saberhagen, "Cryptonote," 2013, https://cryptonote.org/whitepaper.

[129] T. Jedusor, "Mimblewimble," 2016, https://scalingbitcoin.org/papers/mimblewimble.txt.

[130] A. Poelstra, "Mimblewimble," 2016, http://diyhpl.us/wiki/transcripts/scalingbitcoin/milan/mimblewimble/.

[131] E. K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, "Enhancing bitcoin security and performance with strong consistency via collective signing," in *25th USENIX Security Symposium (USENIX Security 16)*, Austin, TX, 2016, pp. 279–296.

[132] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: Generalizing selfish mining and combining with an eclipse attack," in *2016 IEEE European Symposium on Security and Privacy (EuroS P)*, 2016, pp. 305–320.

[133] S. Barber, X. Boyen, E. Shi, and E. Uzun, *Bitter to Better — How to Make Bitcoin a Better Currency*. Springer Berlin Heidelberg, 2012, pp. 399–414.

[134] S. Meiklejohn and C. Orlandi, *Privacy-Enhancing Overlays in Bitcoin*. Springer Berlin Heidelberg, 2015, pp. 127–141.

[135] E. Heilman, F. Baldimtsi, and S. Goldberg, *Blindly Signed Contracts: Anonymous On-Blockchain and Off-Blockchain Bitcoin Transactions*. Springer Berlin Heidelberg, 2016, pp. 43–60.

[136] H. Corrigan-Gibbs and B. Ford, "Dissent: Accountable anonymous group messaging," in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, ser. CCS'10. ACM, 2010, pp. 340–350.

[137] D. Chaum, *Blind Signatures for Untraceable Payments*. Springer US, 1983, pp. 199–203.

[138] A. Kiayias, E. Koutsoupias, M. Kyropoulou, and Y. Tselekounis, "Blockchain mining games," *CoRR*, vol. abs/1607.02420, 2016.

[139] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, and J. S. Rosenschein, "Bitcoin mining pools: A cooperative game theoretic analysis," in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS '15. International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 919–927.

[140] B. A. Fisch, R. Pass, and A. Shelat, "Socially optimal mining pools," *CoRR*, vol. abs/1703.03846, 2017.

[141] A. Kiayias, N. Lamprou, and A.-P. Stouka, *Proofs of Proofs of Work with Sublinear Complexity*. Springer Berlin Heidelberg, 2016, pp. 61–78.

[142] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, Jul. 1982.

[143] "Coindesk. bitcoin venture capital." *Available: http://www.coindesk.com/bitcoin-venture-capital/*.

[144] J. Herrera-Joancomartí and C. Pérez-Solà, *Privacy in Bitcoin Transactions: New Challenges from Blockchain Scalability Solutions*. Springer International Publishing, 2016, pp. 26–44.

[145] Y. Lewenberg, Y. Sompolinsky, and A. Zohar, *Inclusive Block Chain Protocols*. Springer Berlin Heidelberg, 2015, pp. 528–547.

[146] G. Ateniese, B. Magri, D. Venturi, and E. Andrade, "Redactable blockchain – or – rewriting history in bitcoin and friends," 2016, http://eprint.iacr.org/2016/757.

[147] M. Rosenfeld, "Analysis of hashrate-based double spending," *CoRR*, vol. abs/1402.2009, 2014.

[148] M. Milutinovic, W. He, H. Wu, and M. Kanwal, "Proof of luck: an efficient blockchain consensus protocol," *CoRR*, vol. abs/1703.05435, 2017.

[149] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *IEEE Symposium on Security and Privacy*, May 2016, pp. 839–858.

[150] G. Zyskind, O. Nathan, and A. Pentland, "Enigma: Decentralized computation platform with guaranteed privacy," *CoRR*, vol. abs/1506.03471, 2015.