This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

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 proof-of-work (PoW) based distributed consensus protocol, which is run by the network nodes called *miners*. In exchange for the incentive, the miners are expected to maintain the blockchain honestly. Since its launch in 2009, Bitcoin economy has grown at an enormous rate, and it is now worth about 150 billions of dollars. This exponential growth in the market value of bitcoins motivate adversaries to exploit weaknesses for profit, and researchers to discover new vulnerabilities in the system, propose countermeasures, and predict upcoming trends.**

In this paper, we present a systematic survey that covers the security and privacy aspects of Bitcoin. We start by giving an overview of the Bitcoin system and its major components along with their functionality and interactions within the system. We review the existing vulnerabilities in Bitcoin and its major underlying technologies such as blockchain and PoW-based consensus protocol. These vulnerabilities lead to the execution of various security threats to the standard functionality of Bitcoin. We then investigate the feasibility and robustness of the state-of-the-art security solutions. Additionally, we discuss the current anonymity considerations in Bitcoin and the privacy-related threats to Bitcoin users along with the analysis of the existing privacy-preserving solutions. Finally, we summarize the critical open challenges, and we suggest directions for future research towards provisioning stringent security and privacy solutions for Bitcoin.

***Keywords*—Bitcoins, cryptocurrency, security threats, user privacy**

I. Introduction

BITCOIN uses peer-to-peer (P2P) technology, and it operates without any trusted third party authority that may appear as a bank, a Chartered Accountant (CA), a notary, or any other centralized service [1]. An owner has full control over its bitcoins, and she could spend them anytime and anywhere without involving any centralized authority. Bitcoin design is open-source and nobody owns or controls it. Moreover, Bitcoin is a cryptographically secure electronic payment system, and it enables transactions involving virtual currency in the form of digital tokens called Bitcoin coins (BTC or simply bitcoins).

Since its deployment in 2009, Bitcoin has attracted a lot of attention from both academia and industry. With a market capitalization of about 150 billion and more than 150,000 aggregate number of confirmed transactions per day (as of April 2018), Bitcoin is the most successful cryptocurrency to date. Given the amount of money at stake, Bitcoin is an obvious target for adversaries. Indeed, numerous attacks have been described targeting different aspects of the Bitcoin system, these attacks includes double spending [2], netsplit [3], transaction malleability [4], networking attacks [5], and attacks targeting mining [6] [7] [8] and mining pools [9]. In [10], authors claim that *Bitcoin works in practice and not in theory* due to the lack of security research to find out theoretical foundation for Bitcoin protocols. Existing security solutions for Bitcoin lacks the required measures that could ensure an adequate level of security for its users. We believe that security solutions should cover all the major components running critical functions in Bitcoin such as blockchain, consensus, key management, and networking protocols. The online communities have already started to use bitcoins with the belief that Bitcoin will soon take over the online trading business. For instance, "Wiki leaks" request its users to donate using the bitcoins. 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 [11].

Recently, Bitcoin technology is grabbing lots of attention from government bodies due to its increasing use by the malicious users to undermine legal controls. In [12], authors call bitcoins *Enigmatic and Controversial Digital Cryptocurrency* due to difficult concepts underneath the Bitcoin system and severe opposition from the government. According to [13], the current bitcoin exchange rate is approximately USD 9000 (as of April 2018) from around USD 1000 at the start of the year 2016. The primary technologies such as blockchain and consensus protocols that makes the Bitcoin a vast success will now also being envisioned in various next-generation applications. It includes smart trading in smart grids [14], Internet of Things (IoT) [15] [16], vehicular networks [17], health-care data management [18], and smart cities [19], to name a few. As the length of popularity largely depends on the amount of security built on the system which surpasses all

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. He is also Assistant Professor in CSE Department at Manipal University Jaipur, Rajasthan, India e-mail:chhagan@math.unipd.it, ✉Corresponding author

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

its other benefits, we aim to investigate the associated security and privacy issues in Bitcoin and its underlying techniques.

A. Contribution

In this paper, we present a comprehensive survey specifically targeting the security and privacy aspects of Bitcoin and its related concepts. 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 investigate the efficiency of the state-of-the-art security solutions that addresses the existing security and privacy challenges in Bitcoin. In particular, we mainly focus on the security challenges and their countermeasures concerning major components of Bitcoin.

In the literature, [10] provides a discussion on various cryptocurrencies along with a preliminary overview of the advantages and disadvantages of the use of bitcoins. To the best of our knowledge, only the authors in [20] present a comprehensive technical survey on decentralized digital currencies with mainly emphasizing on Bitcoin and its associated technologies. The authors explore the technical background of Bitcoin and discuss the implications of the central design decisions for Bitcoin protocol and its building blocks. However, being a technical survey, authors in [20] are mainly focused on the implementation details of Bitcoin system rather than concentrating on resulting potential issues related to security and privacy of the system as well as of its users. Additionally, [10] and [20] are bit outdated, given the extensive research was done in the last two years on security and privacy, due to increase in the attack vector caused by bitcoins rapid growth in market capitalization and exchange rate. Hence, we firmly believe that a comprehensive survey is essential for an audience who are planning to initiate research in this direction. This paper does not attempt to solve any new challenge but presents an overview and discussion of the Bitcoin security and privacy threats along with their available countermeasures. In particular, the main contributions of our work are as follow.

- We present the essential background knowledge for Bitcoin, its functionalities, and its related concepts. The goal is to enable the new readers to get the required familiarity with the Bitcoin and its underlying technologies such as transactions, blockchain, and consensus protocols. It is notably required to understand, the working methodology, benefits, and challenges associated with the use of bitcoins.
- We systematically present and discuss all the existing security and privacy-related threats that are associated either directly or indirectly (by exploiting one of its underlying technology) with the use of bitcoins. At various levels of its overall operation, we investigate the possibilities, which includes both practical and theoretical risks that an adversary could exploit to launch an attack on the Bitcoin.
- We discuss the efficiency and limitations of the state-of-the-art solutions that address the security threats and enables strong privacy in Bitcoin. Thus, we provide a holistic technical perspective on these challenges in the use of bitcoins. Finally, based on our survey, we give the list of lessons learned, open issues, and directions for future work.

To the best of our knowledge, this is the first survey that discusses and highlights the impact of existing as well as possible future security and privacy threats to Bitcoin and its associated technologies. The paper aims to assist the interested readers: (i) to understand the scope and impact of security and privacy challenges, (ii) to estimate the possible damage caused by these threats, and (iii) to point in the direction that will possibly lead to the detection and containment of the identified risks. In particular, the goal of our research is to raise the awareness in the Bitcoin research community on the pressing requirement to prevent various attacks from disrupting the cryptocurrency. For most of the security threats discussed in this paper, we have no evidence that such attacks have already been performed on Bitcoin. However, we believe that some of the essential characteristics of Bitcoin make these attacks practical and potentially highly disruptive. These characteristics include the high centralization of Bitcoin (from a mining and routing perspective), the lack of authentication and integrity checks for network nodes, and some design choices pertaining, for instance, how in the Bitcoin network a node requests a block.

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 many security threats associated with the development, implementation, and use of various entities of the Bitcoin system. In Section IV, we discuss the state-of-the-art proposals that either countermeasure a security threat or enhances the existing security in Bitcoin. In Section V, we discuss the anonymity and privacy threats towards the use of Bitcoin along with their current solutions. We present the summary of the observations and future research directions that are learned from our survey 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]. The Bitcoin nodes communicate using a peer-to-peer (P2P) network. To achieve consensus among nodes, Bitcoin uses a probabilistic distributed consensus protocol. In Bitcoin, electronic payments are done by generating transactions that transfer *bitcoins* among users. The destination address (also called *Bitcoin address*) is generated by performing a series of irreversible cryptographic hashing operations on the user's public key. In Bitcoin, a user can have multiple addresses by creating numerous public keys, and these addresses could be associated with one or more of her wallets [21]. The private key of the user is required to spend the owned bitcoins in the form of digitally signed transactions. Using the hash of the public key as a receiving address provides the users a certain degree of anonymity (i.e.,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

pseudonymity), and it is recommended the practice to use different Bitcoin address for each receiving transaction.

In Bitcoin, transactions are processed to verify their integrity, authenticity, and correctness by a group of resourceful network nodes called "Miners". In particular, instead of mining a single transaction, the miners bundle some transactions that are waiting for the network to get processed in a single unit called "block". The miner advertises a block in the whole network as soon as it completes its processing (or validation/mining) to claim the mining reward. The newly mined block is verified by the majority of miners in the network before it is successfully added in a distributed public ledger called "blockchain". The miner who mines a block receives a reward once the mined block is successfully added in the blockchain. We now present an overview of the significant technical components and operational features that are essential for the practical realization of the Bitcoin system.

A. Transaction and Proof-of-Work

Bitcoin use transactions to move coins from one user wallet to another. In particular, the coins are represented in the form of transactions, more specifically, a chain of transactions. As depicted in Figure 1, the key fields in a transaction includes Bitcoin version, a hash of the transaction, *Locktime*[1], one or more inputs, and one or more outputs. Every input in a transaction belongs to a particular user, and it consists of the following: (i) hash pointer to a previous transaction which serves as the identifier of the transaction that includes the output, which we now want to utilise as an input in the current transaction, (ii) an index to specific unspent previous transaction output (UTXO) that we want to spend in the current transaction, (iii) *unlocking* script length, and (iv) *unlocking* script (also referred to as scriptSig) which satisfies the conditions associated with the use of UTXO. A transaction output consists of, the number of bitcoins that are being transferred, *locking* script length, and *locking* script (also referred to as scriptPubKey) which imposes a condition that must be met before the UTXO can be spent. To authorize a transaction input, the corresponding user of the input provides the public key and the cryptographic signature generated using her private key. Multiple inputs coming from various previous transactions are often listed in a single transaction. All these previous transactions input values are added up, and the total (excluding transaction fee, if any) is completely used by the outputs of the current transaction.

When the output of a previous transaction is used as the input in a new transaction, it must be spent in its entirety. Sometimes the coin value of the output is higher than what the user wishes to pay. In this case, the sender generates a new Bitcoin address and sends the difference back to this address. For instance, Bob has 50 coins from one of its previous transaction's output, and he wants to transfer 5 coins to $Alice$ using that output as an input in a new transaction. In this particular case, Bob has to create a new transaction with one input (i.e., output from its previous transaction where Bob has received the 50 bitcoins) and two outputs. In the outputs, one output will show that 5 coins are being transferred to Alice, and other output will show a transfer of the remaining coins to one of the wallet owned by Bob. With this approach, the Bitcoin achieves two goals: (i) it implements the idea of $change$, and (ii) one can quickly identify the unspent coins or balance of a user by only looking the outputs from its previous transactions. 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 bitcoins. 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 the other script is called "Pay-to-ScriptHash" (P2SH) [22], which is typically used as multi-signature addresses, but it also enables a variety of transaction types and supports future developments.

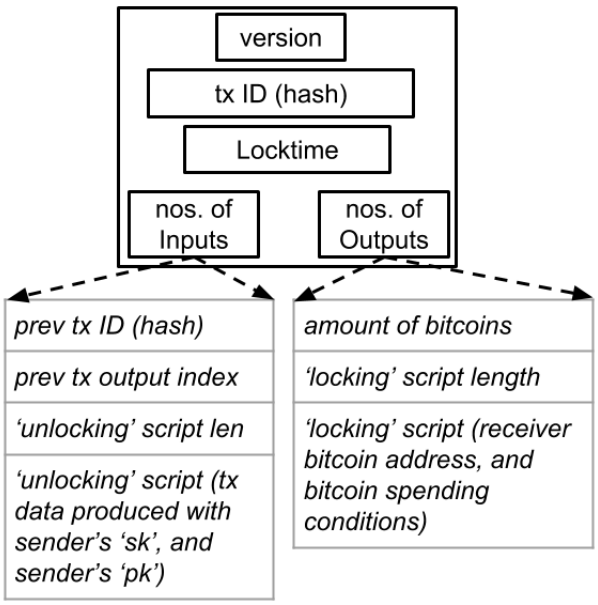


Fig. 1. Bitcoin transactions

In central bank, all the transactions are verified, processed, and recorded in a centralized private ledger. While in Bitcoin, every user acts as a bank and it keeps a copy of the ledger. In Bitcoin, the role of the distributed ledger is given to the so-called *blockchain*. Due to the storage of multiple copies of blockchain at multiple nodes in the network, new vulnerabilities arises such as keeping a consistent global view of the blockchain in the whole system. For instance, a user (say $Alice$) could simultaneously generate two different transactions, using the same set of coins, to two different receivers (say, Bob and $Carol$). This type of malicious behavior by a user is known as *double spending*. Now, if both the receiver processes the transaction independently based on their local view of the blockchain, and the transaction verification is successful, then it leaves the blockchain into an inconsistent state. The primary requirements to avoid the

[1]It indicates the earliest time or blockchain length when this transaction may be spent to the blockchain.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

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 requirements mentioned above, Bitcoin uses the concept of *Proof-of-Work* (PoW) and a probabilistic distributed *consensus protocol*.

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. 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, in this way the correct state of the blockchain is obtained by election. However, this scheme is vulnerable to the sybil attacks [23]. With sybil attack, a miner creates multiple virtual nodes in the 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 based consensus model, in which to verify a transaction, the miners have to perform some computational task to prove that beyond their virtual representatives there are real entities. The PoW consists of a complex cryptographic math puzzle, similar to Adam Back's *Hashcash* [24]. In particular, PoW involves scanning for a value (called *nonce*) that when hashed such as with SHA-256, the resulting hash begins with the required number of zero bits. The number of zero bits required is set by the *target*. The resulting hash has to be a value less than the current *target* and so it must consist of a certain number of leading zero bits to be less than that. In this way, PoW imposes a high level of computational cost on the transaction verification process, and the verification will depend 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 than it is to perform a sybil attack in the network.

In practice, the miners do not mine individual transactions. Instead, they collect pending transactions to form a *block*. The miners mine a block by calculating the hash of that block along with a varying nonce. The nonce is varied until the resultant hash value becomes lower or equal to a given *target* value. The *target* is a 256-bit number that all miners share. Calculating the desired hash value is computationally challenging. For hashing, Bitcoin uses SHA-256 hash function [25]. Unless the cryptographic hash function finds the required hash value, the only option is to try different nonce until a solution (a hash value lower than the target) is discovered. Consequently, the difficulty of the puzzle depends on the target value, i.e., lower the target, the fewer solutions exist, hence more difficult the hash calculations become. Once a miner calculates the correct hash value for a block, it immediately broadcast the block in the network along with the calculated hash value and nonce, and it append the block in its private blockchain. The other miners upon reception of a mined block can quickly verify its correctness by just comparing the hash value given in the received block with the *target* value. The other 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 (as of April 2018, 12.5 BTCs) with a set of newly generated coins. This reward halves every 210,000 blocks. The mining rewards are not received from anyone because there is no central authority to do it. Instead, the rewards are kept part of the block generation process, in which a miner inserts a *reward generating transaction* (or a coinbase transaction) for its Bitcoin address, and it is always the first transaction appearing in every block. If the mined block is validated and accepted by the peers, then this inserted transaction becomes valid, and the miner receives the rewarded bitcoins.

All the miners race to calculate the correct hash value for a block by performing the PoW so that they can collect the associated reward. The chance of being the first to solve the puzzle is higher for the miners who own or control 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 reasonable waiting time for the block validation and generation, the $target$ value is adjusted after every 2,016 blocks. This adjustment of the $target$ also helps in keeping per block verification time to approximately 10 minutes. It further affects the new bitcoins generation rate by keeping the block verification time to approximately 10 minutes. It implies that, on average, nearly 12.5 new bitcoins can be added to the network per 10 minutes (as of April 2018).

Apart from the mining reward, for every successful addition of a transaction in the blockchain, the miner will also receive an amount called *transaction fee*, which is equivalent to the amount remaining when the value of all outputs in a transaction is subtracted from all its inputs [26]. As the mining reward keeps on decreasing with time, and the number of transactions are rapidly increasing in the network, the transaction fee takes a significant role for how fast a transaction is to be included in the blockchain. Authors in [27] investigate the evolution of transactions fees to understand its role and behavior along with its influence on the dynamics and stability of the blockchain. The investigation shows that if the arrival rate of potential transactions at miners is low, transactions without (or very low) fees attached were written to the blockchain, but as the arrival rate of new transactions increases, the transactions with higher fees attached are posted first to the blockchain. The research also shows that higher transactions fees are being driven by queuing problems facing users, rather than by reductions in block rewards. In particular, a transaction with low transaction fee could suffer from the *starvation* problem, i.e., denied service for a long time, if the miners are busy processing the transactions with a higher transaction fee. The Bitcoin never mandates transaction fee, and it is only specified by the owner(s) of a transaction, and it is different for each transaction. However, as users battle to get transactions posted on the blockchain, the transaction fees are rising to levels that discourage bitcoin usage, highlighting a significant structural issue confronting the blockchain [28]. Authors in [29] investigates the possible security issues in Bitcoin that might arise in the absence of block rewards. The main focus in [29] is to analyze deviant mining strategies (such

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

as selfish mining) in the transaction fee regime that can harm Bitcoin security.

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 form of $blocks$. In each block, the transactions are stored using Merkle Tree [30], and a relatively secure time-stamp and a hash of the previous block is also stored. Figure 2 shows the working methodology that is used for creating and maintaining the blockchain. To successfully add a new block in the blockchain, the miners need to verify (mine) a block by solving a computationally difficult PoW puzzle. One can traverse the blockchain to determine the ownership of each bitcoin because the blocks are stored in an ordered fashion. However, tempering within a block is not possible as it would change the hash of the block. If a transaction in a block is tampered, the hash value of that block will change, and it will change the subsequent blocks because each block contains the hash of the previous block. The blockchain continually 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 the distribution of a verified block is delayed due to network latency, both incidents result a valid blockchain $forks$ of equal length. Although, the forks are undesirable as the miners need to keep a global state of the blockchain that consists 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 when miners operating on one fork will broadcast a valid block before the others. Therefore, a longer version of the blockchain now exists in the network, and all the miners will start adding their following blocks on top of the longer blockchain. The aforementioned behavior of blockchain is shown in Figure 3.

The presence of blockchain forks in Bitcoin could be exploited by a malicious miner to gain profits or to disturb the normal functioning of the Bitcoin system. A resourceful miner (or mining pool) could force a blockchain fork in the network by privately mining on it. Once the malicious miner sees that the length of the public blockchain is catching up fast with her private chain, the miner broadcast her blockchain in the network, and due to its longer length, all the other miners will start mining on top of it. In this process, all the mined (i.e., valid) blocks on the separate 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 that exploits the forking nature of blockchain.

In general, the security in Bitcoin is on the assumption that the honest players control a majority of the computing resources. The primary driving factor for miners to honestly verify a block is the reward (i.e., 12.5 BTCs) that they receive upon every successful block addition in the blockchain. As mentioned before that to verify a block the miners need to solve the associated hard crypto-puzzle. The probability of solving the crypto-puzzle is proportional to the number of computing resources used. As per [31], a single home miner who 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 successful mining, the manager distributes the reward among 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 [32] [33].

For the better understanding how a transaction is being processed in the Bitcoin, please refer to Figure 4. Assume that Bob wants to transfer 5 bitcoins to $Alice$. To pay $Alice$, Bob needs a device such as a smartphone, tablet, or laptop that runs the Bitcoin full or lightweight client-side software, and two pieces of information which include $Bob's$ private key and $Alice's$ Bitcoin address. Any user in the network can send money to a Bitcoin address, but only a unique signature generated using the private key can release bitcoins from the account. Bob uses a cryptographic key to digitally sign off on the transaction, proving that he owns those coins. When Bob broadcast a transaction in the network, an alert is sent to all the miners in the network informing them about this new transaction. The miners check that the digital signatures are correct, and Bob has enough bitcoins to complete the transactions. Additionally, miners race to bundle all the pending transactions (including $bob's$) in the network and mine the resulting block by varying the nonce. In particular, the miners create a hash of the block, and if the hash does not begin with a specified number of zeros, the hash function is rerun using a new nonce. The miner randomly selects an initial value for the nonce and increment it for each rerun till either the puzzle is solved, or some other miner solves it. The required hash value must have an absolute but arbitrary number of zeros at the beginning. It is unpredictable which nonce will generate the needed hash with a correct number of zeros, so the miners have to keep trying by using different nonce to find the desired hash 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. Both, the Bob and the $Alice$ will also receive a confirmation about the successful transaction. Other miners communicate their acceptance, and they turn their attention to discover the next block in the network. However, a successful transaction could be discarded or deemed invalid at later period 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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

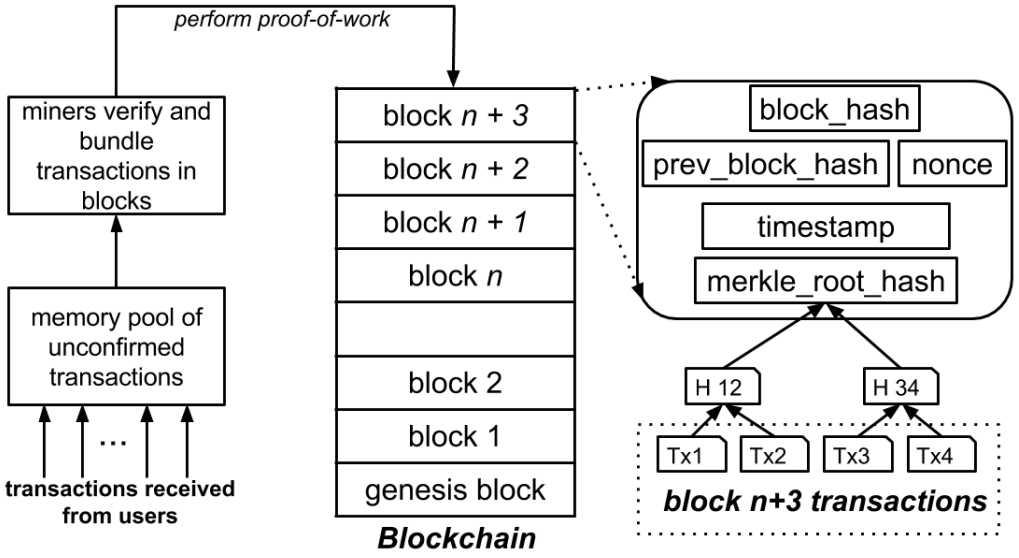


Fig. 2. Creation and addition of blocks in blockchain

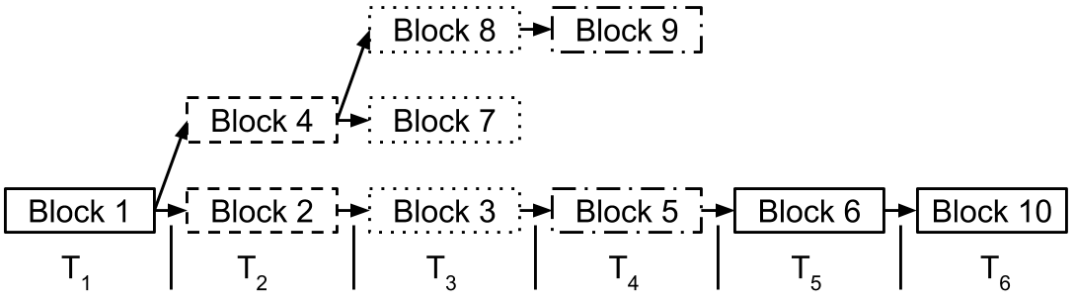


Fig. 3. Blockchain consensus model

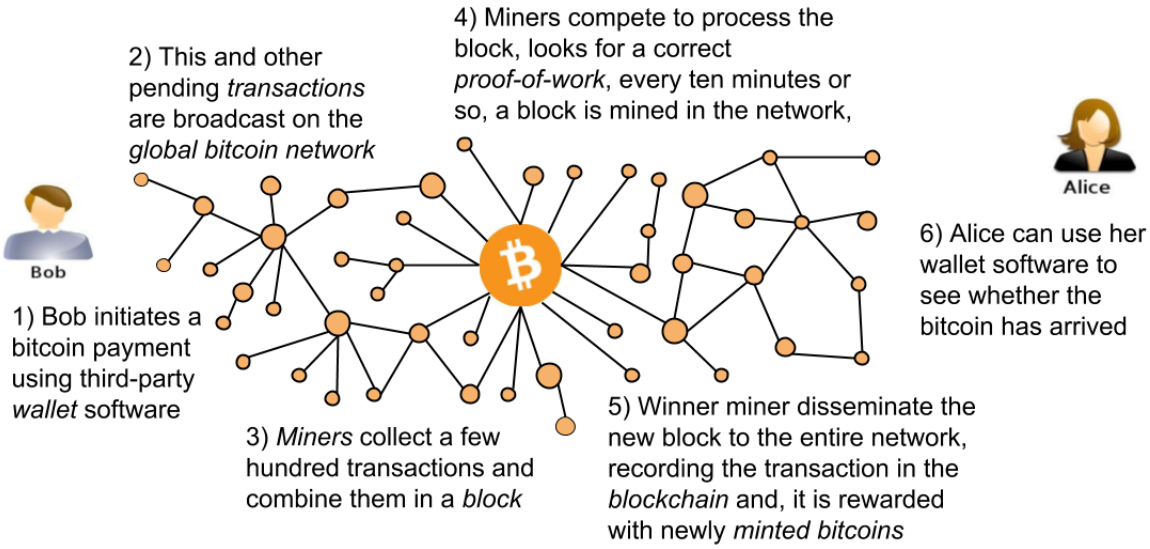


Fig. 4. Bitcoin transaction processing steps

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

as a valid block, a double spending attack is detected, to name a few. The Bitcoin protocol rewards the winning miner with the set of newly minted bitcoins as $incentive$, and the hashed block is published in the public ledger. Once $Bob's$ transaction has been added in the blockchain, he and $Alice$ each receive the first confirmation stating that the bitcoins has been signed over to $Alice$. Concerning transaction time, it depends on the current network load and the transaction fee included in the transaction by Bob, but at the minimum, it would be around 10 minutes. However, receiving the first confirmation does not mean that the transaction is processed successfully, and it cannot be invalidated at a later point in 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 six confirmations) before it is considered as a valid transaction.

C. Consensus Protocol

Bitcoin blockchain is a decentralized system. Thus, it does not require authorization from any trusted third party (TTP) to process the transactions. In particular, the nodes communicate over a network and collaboratively construct the blockchain without relying on a central authority. However, individual nodes might crash, behave maliciously, act against the common goal, or the network communication may become interrupted. For delivering a continuous service, the network, therefore, run a fault-tolerant consensus protocol to ensure that all nodes agree on the order in which entries are appended to the blockchain. To add a new block in the blockchain, every miner must follow a set of rules specified in the consensus protocol. Bitcoin achieves the distributed consensus by using PoW based consensus algorithm. The algorithm imposes the following major rules: (i) input and output values are rational, (ii) transactions only spend unspent outputs, (iii) all inputs being spent have valid signatures, (iv) no coinbase[2] transaction outputs were spent within 100 blocks of their creation, and (v) no transaction spend inputs with a locktime before the block in which they are confirmed. Generally, a blockchain based system such as Bitcoin is considered as secure and robust as its consensus model.

In the PoW based consensus algorithm, the participants require no authentication to join the network. It makes the Bitcoin consensus model extremely scalable regarding supporting thousands of network nodes. However, PoW based consensus is vulnerable to "51%" attacks, in which an adversary has control over 51% of the mining power (i.e., hash rate) in the network. Hence it can write its blocks or fork the blockchain that at a later point converges with the main blockchain. Such malicious behavior of an adversary also helps her to perform several other types of attacks such as double spending, eclipse, and denial-of-service. In particular, 51% attack drives away the honest miners from the mining process, thus it weakens the consensus protocol which poses a threat to Bitcoin security and robustness. One way to achieve the 51% attack is to incentivize (or bribe) the honest miners to join the attackers' coalition.

[2]A coinbase transaction is a unique type of bitcoin transaction that can only be created by a miner.

It is vital for Bitcoin to have a broadcast network which is not only decentralized but it also provides low latency, and it is difficult to deliberately censor or delay messages. Although the PoW based consensus algorithm wastes a lot of energy in hash computations during the mining process, it facilitates high scalability concerning number of nodes participating in the network, and it operates entirely in a decentralized fashion. Along with the various security attacks (please refer to tables I and II), the effectiveness of a consensus protocol also depends on the performance and stability of the network. For instance, an increase in the latency between the validation of a block and its receipt by all other miners increases the possibility of a temporary blockchain fork. Although, due to the PoW model eventual consistency in the blockchain will be reached despite the temporary forks, however, it results in longer transaction confirmation times. Bitcoin transactions have an average size of 250 bytes, and the maximum block size was set at 1 MB which can contain approximately 4,000 transactions. Blocks are mined on average every 10 minutes, which implies an average transaction rate of some 7 transactions per second (tps). The block size limitation of Bitcoin is increasingly felt by its users in the form of delayed transaction processing and rising transaction fees, thus it has given rise to several proposals about how to increase the transaction throughput [34]. Currently, the Bitcoin is not at par with VISA which has an average transaction rate of some 2,000 transactions per second (tps) [35] with a peak of 56,000 tps. Bitcoin is also not at par with PayPal which has an average transaction rate of some 170 tps [36]. Furthermore, the transaction fee could continue to grow as competition for space in the blockchain increases, and the protocol's monetary policy continually reduces the minting of new coins that rewards miners for securing the network.

The Bitcoin tps can be increased by increasing the block size and/or by decreasing the block discovery interval. Both of these interventions will increase the end-to-end block transmission delay, which in turn will increase the probability that different participants momentarily record different versions of the blockchain, so that the consensus protocol will discard an increasing number of blocks. The net effect is that the real increase in the tps is not proportional to the increase (decrease) in the block size (block discovery rate) [37].

As an attempt to increase the scalability and tps in Bitcoin, new proposals such as Segregated Witness (*SegWit*) [38] and *micropayment channel networks* (e.g., Bitcoin Lightning networks) [39] [40] have been proposed. SegWit (BIP141) was an attempt to increase the number of transactions within a block while keeping the same (i.e., 1MB) block size limit. It was activated successfully on Bitcoin on 23 August 2017. SegWit splits the transaction into two segments. It is done by removing the unlocking signature (*witness* data) from the original portion and appending it as a separate structure at the end. In particular, once SegWit is activated, all the sender and receiver details will go inside the main block, and a new *witness* structure would contain scripts and signatures. In this way, the SegWit creates more space in the blocks to fit the additional number of transactions. SegWit should not be confused with SegWit2x, which proposed to first activate SegWit and then a 2MB hard fork. SegWit2x was not activated, as it did not gain consensus.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

As a result of this, the Bitcoin Cash (BCC) was formed, which does not have SegWit but increased the block size to 8MB. Specifically, the Bitcoin Cash is a hard fork of the Bitcoin network which chose to implement a larger block size limit (i.e., 8MB) rather than rely on a new transaction structure, e.g., SigWit[3]. Bitcoin Cash was the result of enforcing BIP91 [41] on Bitcoin, and it was activated successfully on 1 August 2017. At that time, the users who possessed BTC were immediately granted an equivalent amount of BCC against the BTC they possessed. Although, the Bitcoin Cash has few advantages over Bitcoin such as increased tps, resistance against replay attacks, and network scalability, but from security viewpoint, having fewer validators in Bitcoin Cash due to larger block size will imply fewer individuals ensuring ledger accuracy. Additionally, it also results in fewer entities that would be able to validate the blockchain as part of the mining process, which results in encouraging miner centralization.

To address scalability and tps while keeping the block size unchanged, micropayment channel networks have been proposed [42] [39]. In these networks, a payment channel is established between two parties, who make payments to one another, none of which are recorded on the blockchain. This off-chain mode of payment helps to process payments faster. In particular, a micropayment channel provides a way to trustlessly track money transfers between two entities off-blockchain with smart contracts. For instance, authors in [39] propose a decentralized system in which transactions are sent over a network of micropayment channels whose transfer of value occurs off-blockchain. However, these payment channel networks cause a new set of challenges related to routing [43], processing concurrent payments [44], and user privacy [45] [46].

Bitcoin consensus algorithm has been its most widely debated component in the Bitcoin research community. It is because the consensus algorithm rises: (i) open questions about the Bitcoin stability [10]; (ii) concerns about the performance and scalability of the protocol [47]; and (iii) concerns for computational resource wastage [48]. In particular, the PoW consensus model used by the blockchain is very inefficient concerning the power consumption and the overall generation time of new blocks. Hence, to overcome or limit some of the aforementioned disadvantages of PoW, various other consensus protocols such as Proof-of-Stake (PoS) [49], Proof of Elapsed Time (PoET), Proof of Authority (PoA), Federated Byzantine Fault Tolerance (FBFT), Proof of Storage [50] [51], to name a few are designed. The most obvious difference between these consensus protocols and PoW is that in each of these alternative protocols, the consensus is driven at the expense of internal resources (e.g., coins or reputation) instead of external resources (e.g., electricity). This creates an entirely different set of incentives for (and trust in) network nodes (i.e., miners), which drastically changes the network security model. Detailed discussions on these alternative consensus protocols are out of the scope of our survey. Hence we direct interested users to [52] [20] [53] [54] [55].

[3]https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki

D. Networking Infrastructure

Bitcoin uses an unstructured peer-to-peer (P2P) network [56] based on the non-encrypted persistent TCP connections as its foundation communication structure. An unstructured P2P network organizes the peers in a random graph by following a flat or hierarchical organization. It utilizes flooding and similar opportunistic techniques such as random walks, expanding-ring, Time-to-Live (TTL) search etc., to locate peers that have interesting data items. In general, unstructured overlays are easily constructed and are robust against highly dynamic network topologies, i.e., against frequently joining and leaving peers. These type of networks are best suited for Bitcoin as the aim is to distribute information as fast as possible to reach consensus on the blockchain. However, experimenting with the Bitcoin network/protocol poses a challenge. By now, there are a few possibilities to approach this task. One way is to connect to the mainnet, i.e., the live Bitcoin network, or the testnet. Another way is to use the simulation environments such as Shadow [57] event discrete simulator, which aims at simulating large-scale Bitcoin networks, while keeping full control over all components.

Bitcoin nodes maintain a list of IP addresses of potential peers, and the list is bootstrapped via a DNS server and additional addresses are exchanged between peers. Each peer aims to maintain a minimum of 8 non-encrypted TCP connections in the overlay, i.e., the peer actively tries to establish additional connections if the current number of connections is lower than 8. The number of eight connections can be significantly exceeded if incoming connections are accepted by a Bitcoin peer up to a maximum of 125 connections at a time. By default, peers listen on port 8333 for inbound connections. When peers establish a new connection, they perform an application layer handshake, consisting of version and verack messages. The messages include a timestamp for time synchronization, IP addresses, and the protocol version. A node select its peers in a random fashion, and it picks a new set of peers after a fixed amount of time. This is done to minimize the possibility and effects of *netsplit* attack, in which an attacker creates an inconsistent view of the network (and the blockchain) at the attacked node. Since Bitcoin version 0.7, IPv6 is supported. To detect when peers have left, Bitcoin uses a soft-state approach. If 30 minutes have been passed since messages were last exchanged between neighbors, peers will transmit a hello message to keep the connection alive.

Miners continually listen to new block announcements which are sent via INV messages containing the hash of the mined block. If a miner discovers that it does not hold a newly announced block, it transmits a $GETDATA$ message to one of its neighbor. The neighbor then responds by sending the requested information in a $BLOCK$ message. In case the requested block do not arrive within 20 minutes, the miner triggers the disconnection of that particular neighbor and asks the same information from another neighbor. The propagation of transactions occurs in a sequence given as INV, $GETDATA$, and TX messages, in which nodes announce request and share transactions that have not yet been included in the blockchain. To form the distributed consensus, newly

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

discovered transactions and blocks are propagated (through flooding) in the whole network. Miners store new transactions for the mining purposes, but after some time remove them if they do not make it on the blockchain. It is the responsibility of the transaction originator that the transaction is received by all the peers in the network. To this end, the originator might need to rebroadcast the transaction if it did not get into the blockchain in the first attempt. It is to ensure that the transaction gets considered in the next block.

To maintain the consistent globe view of blockchain at the network nodes and to avoid blockchain forking, Bitcoin requires quick dissemination of newly generated transactions and mined blocks. However, this requirement exposes the Bitcoin networking infrastructure to an array of routing attacks. For instance, Bitcoin connections are routed in the form of clear text and without any integrity checks, which means that any adversary on the forwarding path can discard, eavesdrop, modify or insert Bitcoin messages (e.g., blocks and transactions). Additionally, an adversary could introduce delay in the propagation of both, new transactions and mined block, to launch the double spend and netsplit attacks. As shown in [58], the propagation time can even be further extended under reasonable circumstances. Authors in [5] present a taxonomy of routing attacks and their impact on Bitcoin, considering both small-scale attacks, targeting individual nodes, and large-scale attacks, targeting the network as a whole. It is quite possible that by isolating parts of the network or delaying block propagation, adversaries could cause a significant amount of mining power to be wasted, leading to revenue losses and exposing the network to a wide range of exploits. Detecting and mitigating these network attacks is a challenging task due to following two reasons, first it requires inferring the exact forwarding paths taken by the Bitcoin traffic using measurements (e.g., traceroute) or routing data (BGP announcements), both of which can be forged, and second it is essentially a human-driven process consisting of filtering, routing around or disconnecting the attacker.

The use of an unstructured P2P network in Bitcoin enables the required rapid distribution of information in every part of the network. The security of Bitcoin heavily depends on the consistent global state of blockchain which relies on the efficiency of its PoW based consensus protocol. The variations in the propagation mechanisms could adversely affect the consensus protocol. The presence of inconsistent blockchain states, if exploited correctly could lead to successful double spending. To this end, it is essential that the Bitcoin network should remain scalable concerning network bandwidth, network size, and storage requirements because this will facilitate the increase in the number of honest miners in the network, which will strengthen the consensus protocol. In Bitcoin, full nodes download and verify all blocks starting from the genesis block because it is the most secure way. Full nodes participate in the P2P network and help to propagate information, although it is not mandatory to do so. Alternatively, the thin clients use the simplified payment verification (SPV) to perform transactions. The SPV is a method used by Bitcoin thin client for verifying if particular transactions are included in a block without downloading the entire blockchain. More specifically, during the initial syncing process, a thin client only downloads a copy of the block headers of the longest blockchain and then requests transactions from full nodes as needed. However, the use of SPV costs the thin clients because it introduces weaknesses such as Denial of Service (DoS) and privacy leakage for the thin client. In particular, the general scalability issues of unstructured overlays combined with the problems induced by the Bitcoin protocol itself remains in the system. Many of the results suggest that scalability remains an open problem [59] and it is hard to keep the fully decentralized network in future [60] [61].

E. Benefits and Challenges

Same as any other emerging technology, use of Bitcoin comes with certain benefits and challenges, and various types of risks are associated with its use. It is believed that Bitcoin has the following benefits and challenges.

Benefits -

- *No Third-Party Seizure:* No central authority can manipulate or seize the currency since every currency transfer happens peer-to-peer just like hard cash. In particular, bitcoins are yours and only yours, and the central authority cant take your cryptocurrency, because it does not print it, own it, and control it correspondingly.
- *Anonymity and transparency:* Unless Bitcoin users publicize their wallet addresses publicly, it is tough to trace transactions back to them. However, even if the wallet addresses are publicized, a new wallet address can be easily generated. Bitcoin system dramatically increases privacy when compared to traditional currency systems where third parties potentially have access to personal financial data. Moreover, this pseudonymity is achieved without sacrificing the system transparency as all the bitcoin transactions are documented in a public ledger. Unfortunately, numerous research works have shown that the practical technologies of clustering and flow analysis are much effective for tracing Bitcoin transaction and thereby revealing the owner involved [62] [63]. However, to fix the privacy and anonymity flaws in Bitcoin, much work has been done and many schemes proposed in the research community manage to enhance the property of anonymity [64] [65] [66].
- *No taxes and lower transaction fees:* Due to its decentralized nature and pseudonymity, there is no viable way to implement a Bitcoin taxation system. In the past, Bitcoin provided instant transactions at nearly no cost. Even now, Bitcoin has lower transaction costs than a credit card, Paypal, and bank transfers. However, the lower transaction fee is only beneficial in situations where the user performs a substantial value international transactions. This is because the average transaction fee in Bitcoin becomes higher for minimal value transfers or purchases such as paying for regular household commodities.
- *Theft resistance:* Stealing of bitcoins is not possible until the adversary has the private keys (usually kept offline)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

that are associated with the user wallet. In particular, Bitcoin provides security by design, for instance, unlike with credit cards you dont expose your secret (private key) whenever you make a transaction. Moreover, bitcoins are free from *Charge-backs*, i.e., once bitcoins are sent, the transaction cannot be reversed. Since the ownership address of the sent bitcoins will be changed to the new owner, and it is impossible to revert. This ensures that there is no risk involved when receiving bitcoins.

Challenges:

- *High energy consumption:* Bitcoin blockchain uses PoW model to achieve distributed consensus in the network. Although the use of PoW makes the mining process more resistant to various security threats such as sybil and double spending, it consumes a ridiculous amount of energy and computing resources [67] [68]. As mentioned in Section II-B that the miners bundle a set of transactions to create a block and to mine the block, it is hashed by varying the nonce. However, the hashing is not inherently computationally intensive, but to get the required hash that starts with the required number of zeros, a miner has to repeat the hashing process, until the result has the proper number of zeros. This process of hashing and rehashing usually goes on thousands of times, and it is done in parallel in the Bitcoin network by all the miners. Hence it consumes lots of energy. Due to the reason mentioned above, the energy cost for Bitcoin is high in comparison to the conventional financial transactions. For instance, processing a bitcoin transaction consumes more than 5000 times as much energy as using a Visa credit card [69]. Therefore, innovative technologies that reduce the energy consumption are required to ensure a sustainable future of Bitcoin. Furthermore, due to the continuous increase in network load and energy consumption, the time needed for bitcoin transaction processing is increasing.
- *Wallets can be lost:* Since there is no trusted third party if uses lost the private key associated with her wallet due to a hard drive crash or a virus corrupts data or lost the device carrying the key, all the bitcoins in the wallet has been considered lost for *forever*. There is nothing that can be done to recover the bitcoins, and these will be forever orphaned in the system. It can bankrupt a wealthy Bitcoin investor within seconds.
- *(Facilitate) Criminal activity:* The pseudonymity provided by the Bitcoin system helps the would-be cybercriminals to perform various illicit activities such as ransomware [70], tax evasion, underground market, and money laundering. However, the law enforcement could catch the criminals with careful analysis of blockchain data because the transactions are only pseudonymous and the whole history is public. Hence, criminals are starting to use other digital currencies such as Monero or ZCash, which is built specifically for increased user privacy.

According to [71], 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 risks that threaten 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 [72], authors perform a survey on the people's opinion about bitcoins usage. Participants argue that the greatest barrier to the usage of bitcoins 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. Furthermore, 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 [73], 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 bitcoins due to security breaches. Many participants stated that for a fast flow of bitcoins in the user community, an impressive and straightforward user interface is even more important than security. Also, the participants highlighted that the poor usability and lack of knowledge regarding the Bitcoin usage are the major contributors to the security failures.

III. Security: Attacks on Bitcoin Systems

Bitcoin is the most popular cryptocurrency[4] and has stood first in the market capital investment from day one. Since it

[4] www.cryptocoinsnews.com/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

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 and its underlying technologies. 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 in the Bitcoin system. Apart from double spending, which is and will always be possible, 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). Tables I and II provides a comprehensive overview of the potential security threats along with their impacts on various entities in Bitcoin and their possible solutions that exist in literature so far.

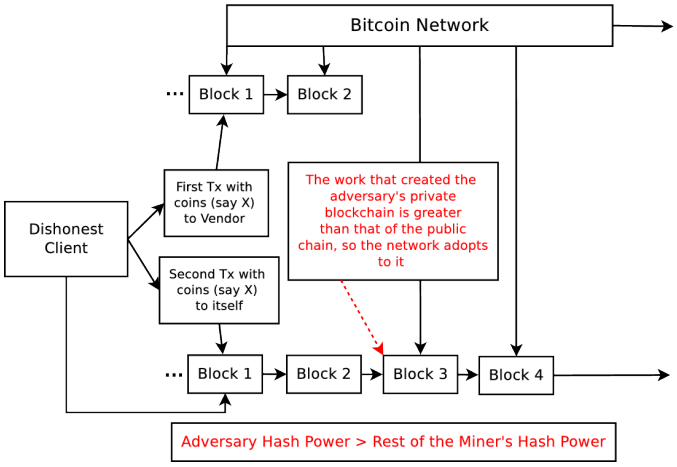


Fig. 5. Double Spending Attack

A. Double Spending

A client in the Bitcoin network achieves a double spend (i.e., send two conflicting transactions in rapid succession) if she can simultaneously spend the same set of bitcoins in two different transactions [2]. For instance, a dishonest client (C_d) creates a transaction $T_V^{C_d}$ at time t using a set of bitcoins (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 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, a *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 the network. The distributed time-stamping and PoW-based 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 5 shows the working methodology of a double spending attack depicting the above explanation.

Despite the use of strict ordering of transactions in the blockchain, PoW scheme, distributed time-stamping [91], and consensus protocol [92] [93], double spending is still possible in Bitcoin. To perform a successful double spending attack, following requirements need to be fulfilled: (i) part of the network miners accept the transaction $T_V^{C_d}$ and the vendor (V) receives the confirmation from these miners, thus releases the product to dishonest client (C_d), (ii) at the same time, other part of the network miners accept the transaction $T_{C_d}^{C_d}$, hence lead to 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 transaction. If the steps mentioned above took place in the given order then the dishonest client can perform a successful double spend. In the rest of this section, we will discuss the variants of double spending attack that are used to realize the aforementioned double spend requirements with varying difficulties and complexities.

A form of double spending called *Finney attack* [77], in which a dishonest client (C_d) pre-mines (i.e., privately) a block that contains the transaction $T_{C_d}^{C_d}$, and then it creates a transaction $T_V^{C_d}$ using the same bitcoins 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}, then as per the Bitcoin protocol rules all the miners in the network will build on top of B'_{fork}. As the B'_{fork} becomes the longest blockchain in the network, all the miners ignore B_{fork}, hence 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, and the client will get back her coins through transaction $T_{C_d}^{C_d}$, but resulting the V losing the product. However, 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* [78] in which a resourceful attacker has control over n nodes in the network, and these nodes collectively work on a private mining scheme with

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

TABLE I. MAJOR ATTACKS ON BITCOIN SYSTEM AND ITS POW BASED CONSENSUS PROTOCOL

Attack	Description	Primary targets	Adverse effects	Possible countermeasures
Double spending [2]	spent the same bitcoins in multiple transactions, send two conflicting transactions in rapid succession	sellers or merchants	sellers lose their products, drive away the honest users, create blockchain forks	inserting observers in network [2], communicating double spending alerts among peers [2], nearby peers should notify the merchant about an ongoing double spend as soon as possible [74], merchants should disable the direct incoming connections [75] [76]
Finney attack [77]	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	wait for multi-confirmations for transactions
Brute force attack [78]	privately mining on blockchain fork to perform double spending	sellers or merchants	facilitates double spending, creates large size blockchain forks	inserting observers in the network [2], notify the merchant about an ongoing double spend [75]
Vector 76 or one-confirmation attack [79]	combination of the double spending and the finney attack	Bitcoin exchange services	facilitates double spending of larger number of bitcoins	wait for multi-confirmations for transactions
$> 50\%$ *hashpower or Goldfinger* [60]	adversary controls more than $> 50\%$ Hashrate	Bitcoin network, miners, Bitcoin exchange centers, and users	drive away the miners working alone or within small mining pools, weakens consensus protocol, DoS	inserting observers in the network [2], communicating double spending alerts among peers [2], disincentivize large mining pools [80] [81], TwinsCoin [82], PieceWork [83]
Block discarding [84] [76] or Selfish mining [6]	abuses Bitcoin forking feature to derive an unfair reward	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 [85] [86], timestamp based techniques such as freshness preferred [87], DECOR+ protocol [88]
Block withholding [31] [89]	miner in a pool submits only PPoWs, but not FPoWs	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 when revenue drops from expected [84], cryptographic commitment schemes [89]
fork after withholding (FAW) attack [90]	improves on adverse effects of selfish mining and block withholding attack	honest miners (or mining pools)	waste resources of fellow miners and decreases the pool revenue	no practical defense reported so far

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 attacker can 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 producing 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* [79]. A Bitcoin exchange is a digital marketplace where traders can buy, sell or exchange bitcoins for other assets, such as fiat currencies or altcoins. In this, 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 (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, and it hopes that the exchange or probably some of the nearby miners will consider the blockchain containing the pre-mined block (i..e, B'_{fork}) as the primary chain. The attacker quickly sends another transaction that requests a withdrawal from the exchange of 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 [94] proposes a new attack against the PoW-based consensus mechanism 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 [95] 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) can mine blocks with a faster rate than the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

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 miner. 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 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 likelihood of a double spend. On the other hand, with the increase in the computing resources 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* [60] in which more than 50% computing resources of the network are under the control of a single miner (or mining pool). 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 network.

From the above discussion on the 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, playing 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 bitcoins, 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 to increase the computing power which directly affects the verification time of a block. Hence it increases the chances of winning 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. Generally, mining pools are governed by pool managers which forwards unsolved work units to pool members (i.e., miners). The miners generate *partial proofs-of-work* (PPoWs) and *full proofs-of-work (FPoWs)*, and submit them to the manager as *shares*. Once a miner discovers a new block, it is submitted to the manager along with the FPoW. The manager broadcasts the block in the Bitcoin network to receive the mining reward. The manager distributes the reward to participating miners based on the fraction of shares contributed when compared with the other miners in the pool. Thus, participants are rewarded based on PPoWs, which have absolutely no value in the Bitcoin system. The Bitcoin network currently consists of solo miners, open pools that allow any miner to join, and closed (private) pools that require a private relationship to join.

In recent years, 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. Figure 6 shows the market share till December 2017 of the most popular mining pools. In this section, we will discuss a set of widespread internal and external attacks on the mining pools.

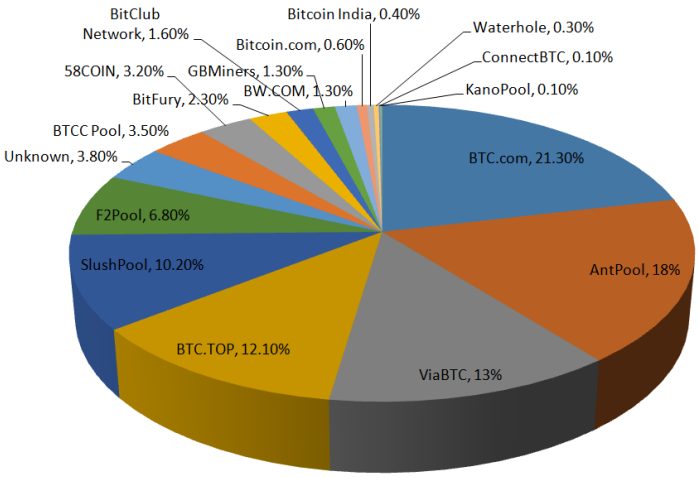


Fig. 6. Bitcoin Hashrate Distribution in Present Market

In a mining pool, the pool manager determines the amount of work done by individual pool members by using the number of shares, a member finds and submit while trying to discover a new block. A *share* is defined as a solution of a crypto-puzzle having a higher target thus a lower number of initial zeros. Moreover, a generic hash is a share with the probability of $1/2^{32}$. Assuming correctness of the hash function used, it is impossible to find shares without doing the work required to discover new blocks or to look for blocks without finding shares along the way. Due to this, the number of shares determined by a miner is proportional, on average, to the number of hashes the miner calculated while attempting to discover a new block for the mining pool. Additionally, in [31], the author discusses the possibility of using variable block rewards and difficulty shares as reward methods in a pool. This variability is introduced due to the following reasons; bitcoins generation per block is cut in half every 210000 blocks, and the transaction fees vary rapidly based on the currently available transactions in the network. As most of the mining pools allow any miner to join them using a public Internet interface, such pools are susceptible to various security threats. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

adversaries believe that it is profitable to *cannibalize* pools than mine honestly. Let's understand it with an example, suppose that an adversary has 30% of hash rate (HR) and 1 BTC is the block mining reward (MR). If the mining pool is sharing the reward based on the invested HR, then the adversary will receive 0.3 BTC for each mined block. Now adversary purchases more mining equipment, worth 1% of current HR. With standard mining strategy, the adversary will gain an additional revenue of 0.0069 BTC for the 1% added HR. By performing pool cannibalizing (i.e., distribute your 1% equally among all other pools, and also withhold the valid blocks) the attacker will still receive the rewards from its pool, but it might also receive additional rewards from the other pools to which she is sharing its 1% HR. This misbehavior will remain undetectable unless the change in reward is statistically significant.

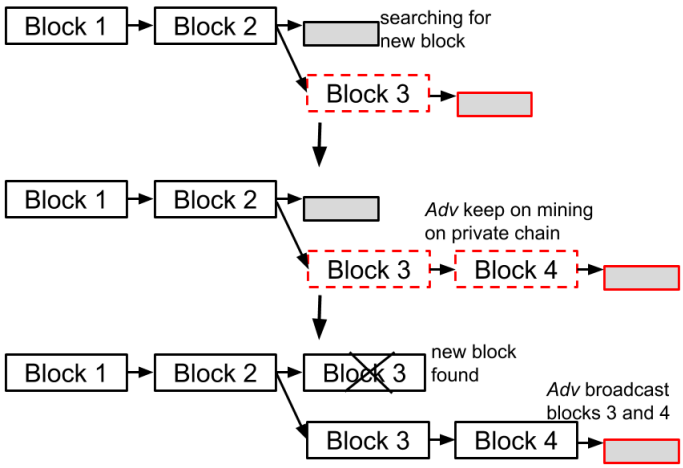


Fig. 7. Selfish Mining

In [84], authors use a game theoretic approach to show that the miners could have a specific sort of subversive mining strategy called *selfish mining* [6] or also popularly known as *block discarding attack* [76] [84]. 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 for the rest of miners, while the *selfish mining* here refers to the malicious miners only. In the selfish mining, 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. As it can be seen in Figure 7 that 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 can take a greater lead on B'_{fork} and they can keep the lead for a longer period, their chances of gaining more reward coins as well as the wastage of honest miners resources increases. To avoid any losses, 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 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 in [6] shows that using the selfish mining, the pool's reward exceed its share of the network's mining power. The statement still holds in cases where the network found their new block before the adversary could find a new second block. Because in such case the miner will make use of the *race to propagate*, i.e., on average the attacker manages to tell 50% of the network about her block first. Additionally, the analysis reveals that the wastage of computing resources and rewards lure honest miners toward the selfish mining pools. Hence it further strengthens the attack. This continuous increase in the selfish pool's size might lead to $> 50\%$ attack, and at that point, the effect of selfish mining will be disastrous.

Another attack much similar to the selfish mining that could be performed on a mining pool is known as *Block withholding* (BWH) [31] [89], in which a pool member never publishes a mined block to sabotage the pool revenue, however, submit shares consists of PPoWs, but not FPoWs. In particular, in [31], two types of block withholding scenarios are presented called "Sabotage" and "Lie in wait". In the first scenario, the adversary does not gain any bitcoins, but it makes other pool members lose, while in the second scenario, the adversary performs a complex block concealing attack similar to the one described in the *selfish mining* attack. In [31], authors discuss a generalized version of the "Sabotage" attack which shows that with slight modification, it is possible for the malicious miner also to earn an additional profit in this scenario. Authors in [48] 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.

To analyze the effects of BWH on mining pools, authors in [9] presents *The Miners Dilemma*, which uses an iterative game to model attack decisions. The game is played between two pools, say pool A and pool B, and each iteration of the game is a case of the *Prisoners Dilemma*, i.e., choose between attacking or not attacking. If pool A chooses to attack pool B, pool A gains revenue, pool A loses revenue, but pool B can latter retaliate by attacking pool A and gaining more revenue. Thus, attacking is the dominant strategy in each iteration, hence if both pool A and pool B attack each other, they will be at a Nash Equilibrium. This implies that if both will earn less than they would have if neither of them attacked. However, if none of the other pools attack, a pool can increase its revenue by attacking the others. Recently, authors in [90] propose a novel attack called a *fork after withholding* (FAW) attack. Authors show that the BWH attackers reward is the lower bound of the FAW attackers, and it is usable up to four times more often per pool than in BWH attack. Moreover, the extra reward for a FAW attack when operating on multiple mining pools is around 56% higher than BWH attack. Furthermore,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

the miners dilemma may not hold under certain circumstances, e.g., when two pools execute FAW attack, the larger pool can consistently win. More importantly, unlike selfish mining, a FAW attack is more practical to execute while using intentional forks.

The *Pool Hopping attack* presented in [31] [96] uses the information about the number of submitted shares in the mining pool to perform the selfish mining. In this attack, the adversary performs continuous analysis of the number of shares submitted by fellow miners to the pool manager to discover a new block. The idea is that if already a large number of shares have been sent and no new block has been found so far, the adversary will be getting a tiny 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.

Recently, the *Bribery attack* is described in [97]. In this, an attacker might obtain the majority of computing resources 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, in which 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 such as double spending and Distributed Denial-of-Service (DDoS) [98]. The miners that took the bribes will get benefits 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.

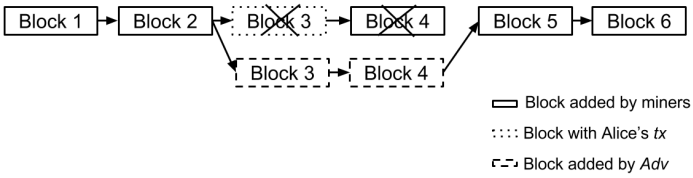


Fig. 8. Blacklisting via Punitive Forking

An adversary with $> 50\%$ hash rate could perform a successful selective blacklisting via *punitive forking*. The objective of punitive forking is to censor the Bitcoin addresses owned by certain people, say *Alice*, and prevent them from spending any of their bitcoins. The strategy to perform the blacklisting (please refer to Figure 8) is as follows: (i) the adversary with $> 50\%$ network hashrate announces to the Bitcion network that she will not extend on the blockchain containing transactions spending from Alice's Bitcoin address, (ii) if some other miners include a transaction from Alice in a block, the adversary will fork and create a longer proof of work blockchain, (iii) Block containing Alice's transaction now invalidated, and it can never be published, also the miner who added the block to Alice's transaction will lose its block reward. However, a weak adversary that has lower hash rate can still cause delays and inconveniences for Alice's transaction.

Punitive forking doesn't work unless you have $> 50\%$ of hash rate. However, there is another strategy to achieve the blacklisting as presented in [99]. In particular, authors present a malicious mining strategy called *feather forking*, in which an attacker announces that she will *attempt* to fork if she sees a block containing Alice's transaction in the blockchain, but she will give up after a while. This is the adversary forks as per its convenience, she will continue to extend its fork until wins (i.e., outraces the main chain), but she gives up (i.e., discard its private fork and continue to extend the main chain) after block with Alice's transaction contains k confirmations. 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 she 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 executing *feather forking* can also use it to *blackmail* a client by threatening that all her transactions will be put on the blacklist until the client pays the asked ransom coins.

C. Client-side Security Threats

The massive 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 to access its account or wallet. Hence, it is desirable to have the key management techniques that are secure, yet usable. This is because 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 the bitcoins, a user needs to install a wallet on 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 [120] for securing the transactions. In particular, Bitcoin uses elliptic curve digital signature algorithm (ECDSA) which is standardized by NIST [121] 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 [122] that even partially bit-wise equal random values suffice to derive a user's

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

TABLE II. MISBEHAVIOR ATTACKS TARGETING BITCOIN NETWORK AND ENTITIES

Attack	Description	Primary targets	Adverse effects	Possible countermeasures
Bribery attacks [97]	adversary bribe miners to mine on her behalf	miners and merchants	increases probability of a double spend or block withholding	increase the rewards for honest miners, make aware the miners to the long-term losses of bribery [97]
Refund attacks [100]	adversary exploits the refund policies of existing payment processors	sellers or merchants, users	merchant losses money while honest users might lose their reputation	publicly verifiable evidence [100]
Punitive and Feather forking [99] [101]	dishonest miners blacklist transactions of specific address	users	freeze the bitcoins of user for forever	remains an open challenge
Transaction malleability [102] [4]	adversary change the TXID without invalidating the transaction	Bitcoin exchange centers	exchanges loss funds due to increase in double deposit or double withdrawal instances	multiple metrics for transaction verification [103], malleability-resilient "refund" transaction [102]
Wallet theft [21]	adversary stole or destroy private key of users	individual users or businesses	bitcoins in the wallet are lost	threshold signature based two-factor security [104] [105], hardware wallets [106], TrustZone-backed Bitcoin wallet [107], Password-Protected Secret Sharing (PPSS) [108]
Time jacking [109]	adversary speed-up the majority of miner's clock	miners	isolate a miner and waste its resources, influence the mining difficulty calculation process	constraint tolerance ranges [109], network time protocol (NTP) or time sampling on the values received from trusted peers [110]
DDoS [111] [112]	a collaborative attack to exhaust network resources	Bitcoin network, businesses, miners, and users	deny services to honest users/miners, isolate or drive away the miners	Proof-of-Activity (PoA) protocol [113], fast verification signature based authentication
Sybil [23]	adversary creates multiple virtual identities	Bitcoin network, miners, users	facilitates time jacking, DDoS, and double spending attacks, threatens user privacy	Xim (a two-party mixing protocol) [114]
Eclipse or netsplit [3]	adversary monopolizes all incoming and outgoing connections of victim	miners, users	inconsistent view of the network and blockchain, enable double spends with more than one confirmation	use whitelists, disabling incoming connections [3]
Tampering [58]	delay the propagation of transactions and blocks to specific nodes	miners, users	mount DoS attacks, wrongfully increase mining advantage, double spend	improve block request management system [58]
Routing attacks [5]	isolate a set of nodes from the Bitcoin network, delaying block propagation	miners, users	denial of service attack, increases possibility of 0-confirmation double spends, increases fork rate, waste the mining power of the pools	increase the diversity of node connections, monitor round-trip time, use gateways in different ASes [5]
Deanonymization [115] [116]	linking IP addresses with a Bitcoin wallet	users	user privacy violation	mixing services [117], CoinJoin [118], CoinShuffle [119]

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 [123]. 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 key. Recently, authors in [124] present a systematic analysis of the effects of broken primitives on Bitcoin. Authors highlight the fact that in the current Bitcoin system has no migration plans in-place for both the broken hash and the broken signature scheme, i.e., the Bitcoin's RIPEMD160, SHA256, and ECDSA techniques are vulnerable to various security threats such as collision attacks [125]. The authors in [124] found that the primary 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 information for user authentication, Bitcoin relies on public key cryptography. This raises the issues of the secure storage and management of the user keys. Over the years, various type of wallet implementations is researched to obtain secure storage of the user keys. It includes software, online or hosted, hardware or offline, paper and brain wallets. Table III shows some popular wallets and their main features. Coinbase (coinbase.com), an online wallet is most popular due to its

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

TABLE III. 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

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 do not have any control over 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 [106] as later are not connected to the Internet, but at the same time hardware wallets lacks usability. If done right, there exist 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 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 risks mentioned above such as managing cryptographic keys [126], lost or stolen devices, equipment failure, Bitcoin-specific malware [127], to name a few, that are associated while storing the bitcoins 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 Bitcoin history is the breakdown and ongoing bankruptcy of the oldest and largest exchange called Mt. Gox, which lost over 450 millions of dollars. Moreover, a few other exchanges have lost their customers bitcoins and declared bankruptcy due to external or internal theft, or technical mistakes [128]. Although, the vulnerability of an exchange system to the disastrous losses can never be fully avoided or mitigated. Therefore the authors in [129] 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. Bitcoin Network Attacks

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) which targets Bitcoin currency exchanges, mining pools, eWallets, and other financial services in Bitcoin. Due to the distributed nature of Bitcoin network and its consensus protocol, launching a DoS attack has no or minimal adverse effect on network functionalities. Hence attackers have to lunch a powerful DDoS to disturb the networking tasks. Unlike DoS attack, in which a single attacker carried out the attack, in DDoS, multiple attackers launch the attack simultaneously. DDoS attacks are inexpensive to carry out, yet quite disruptive. Malicious miners can perform a DDoS (by having access to a distributed Botnet) on competing miners, effectively taking the competing miners out of the network and increasing the malicious miners effective hash rate. In these attacks, the adversary exhausts the network resources to disrupt their access to genuine users. For example, an honest miner is congested with the requests (such as fraudulent 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 [111], 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 massive revenue for the adversary as compared to attacking an individual or small mining pools.

In [112], 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 in increasing its mining capacity 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 the competing mining pools. The majority of the DDoS attacks target large organizations due to bulk ransom motives. Companies like CoinWallet and BitQuick were forced to shut down only after few months of their launch due to the effects of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

continuous DDoS attacks. As stated above that DDoS attack take various forms, one of which is to discourage a miner so that it will withdraw itself from the mining process. For instance, if an attacker displays to a colleague miner that it is more powerful, it can snatch the reward of mining, and it is the apparent winner of the mining process, then honest miner backoff since its chances of winning is less. In this way, an adversary will be successful in removing individual miners as well as small pools from the mining network, thus imposing a DDoS attack on the network [112]. Moreover, in [130], authors propose network partitioning in Bitcoin, hence isolating the honest nodes from the network by reducing their reputation.

Now we discuss the so-called *Malleability attacks* [4], which also facilitates the DDoS attacks in Bitcoin. For instance, by using a *Malleability attack* an adversary clogs the transaction queue [131]. 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. This attack wastes time and resources of the miners and the network [132]. Malleability is defined concerning cryptography by [4]. 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 [102], another form of *malleability* attack called *transaction malleability* is introduced. Suppose that a transaction $T^n_{A\rightarrow B}$ which transfers n bitcoins from $A's$ wallet to $B's$ wallet. With *transaction malleability* it is possible to create another T' that is syntactically different (i.e., $T^n_{A\rightarrow B}$ and T' has different transaction hash ID T^{id}_x) from $T^n_{A\rightarrow B}$, 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 of the success of this attack is that, in Bitcoin each transaction is uniquely identified by its T^{id}_x, hence in some cases T' will be considered a different transaction than $T^n_{A\rightarrow B}$.

In Bitcoin, indeed, 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 mainly more relevant while targeting the Bitcoin exchanges which holds a significant amount of coins. It is because it allows the users to buy and sell bitcoins in exchange for cash or altcoins. The Bitcoin 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 famous exchange) stated 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 C_d can 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.

For the first time, authors in [5] present the impact of routing attacks on Bitcoin network by considering both small and large-scale attacks. The paper shows that two key properties of Bitcoin networks which includes, the ease of routing manipulation, and the rapidly increasing centralization of Bitcoin in terms of mining power and routing, makes the routing attacks practical. More specifically, the critical observations suggest that any adversary with few (< 100) hijacked BGP prefixes could partition nearly 50% of the mining power, even when considering that mining pools are heavily multi-homed. The research also shows that attackers on acting as intermediate nodes can considerably slow down block propagation by interfering with few key Bitcoin messages. Authors back their claims by demonstrating the feasibility of each attack against the deployed Bitcoin software, and quantify their effect on the current Bitcoin topology using data collected from a Bitcoin supernode combined with BGP routing data. Furthermore, to prevent the impact of attacks mentioned above in practice, both short and long-term countermeasures, some of which can be deployed immediately are suggested.

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 [100], 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 bitcoins. Most of the major wallets use BIP70 for bitcoins exchange, and the two dominant Payment Processors called *Coinbase* and *BitPay*, who uses BIP70 and collectively they provide the infrastructure for accepting bitcoins 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 these refund attacks is as follows.

- In *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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

attack by inserting the customers' address as the refund address and send a refund request to the merchant. The merchant sends the amount to the ill merchant, thus 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 herself as a trusted party by making payments through trust-able merchants like CeX. When a customer clicks the webpage, accidentally 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 bitcoins [100].

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* [109]. 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. Hence 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 miners 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 [109]. 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, there are few other minor attacks that we have 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 [23] 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 [133]. 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. Hence it can hamper low latency encryption associated with the 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 [114].

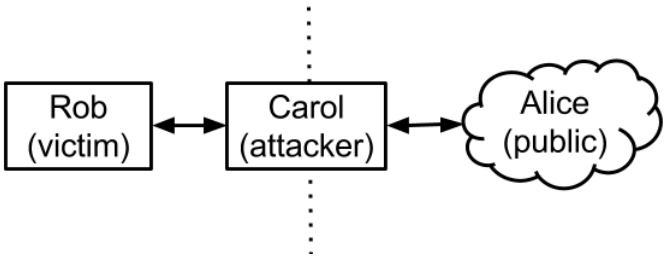


Fig. 9. Eclipse attack

- *Eclipse attack:* In this attack [3], an adversary manipulates a victim peer, and it force network partition (as shown in Figure 9) between the public network and a specific miner (victim). The IP addresses to which the victim user connects are blocked or diverted towards an adversary [3]. Besides, an attacker can hold multiple IP addresses to spoof the victims from the network. An attacker may deploy helpers and launch other attacks on the network such as $N-confirmation$ 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 network with good speed. However, authors in [58] ground this assumption and proved that the adversary could 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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

can become a cause for other types of attacks in the network.

E. Practical attack incidents on Bitcoin exchange systems

In this section, we briefly present the real-world security breaches/incidents that have affected adversely to Bitcoin and its associated technologies, such as blockchain and PoW based consensus protocol. From the start, Bitcoin fans occasionally mentioned about different security threats, typically discussing things like the 51% attack, quantum computer strikes, or an extreme denial of service onslaught from some central bank or government entity. However, these days the word *attack* is used a bit more loosely than ever, as the scaling debate has made people believe almost everything is a Bitcoin network invasion.

One of the biggest attacks in the history of Bitcoin has targeted *Mt. Gox*, the largest Bitcoin exchange, in which a year's long hacking effort to get into Mt. Gox culminated in the loss of 744,408 bitcoins. However, the legitimacy of attack was not completely confirmed, but it was enough to make Mt. Gox shut down and the value of bitcoins to slide to a three-month low. In 2013, another attack called *Silk Road*, the worlds largest online anonymous market famous for its wide collection of illicit drugs and its use of Tor and Bitcoin to protect its user's privacy, reports that it is currently being subjected to what may be the most potent distributed denial-of-service attack against the site to date. In the official statement from the company the following was stated, "The initial investigations indicate that a vendor exploited a recently discovered vulnerability in the Bitcoin protocol known as *transaction malleability* to repeatedly withdraw coins from our system until it was empty". Although transaction malleability is now being addressed by *segwit*, the loss it caused was far too small with the central issue seemingly being at a human level, rather than protocol level. In the same year, *Sheep Marketplace*, one of the leading anonymous websites also announces that they have been hacked by an anonymous vendor *EBOOK101* who stole 5400 bitcoins. However, in all those mentioned above, it remains unclear that whether there is any hacked happened or it is just a fraud by the owners to stole the bitcoins.

Bitstamp, an alternative to MT Gox, increasing its market-share while Gox went under were hacked out of around 5 million dollars in 2015. The theft seems to have been a sophisticated attack, with phishing emails targeting bitstamps personnel. However, as the theft was limited to just hot wallets, they were able to fully cover it, leading to no direct customer losses. *Poloniex* is one of the biggest altcoin exchange with trading volumes of 100,000 BTC or more per day, lost their 12.3% of bitcoins in March 2014. The hack was executed by just clicking withdrawal more than once. As it can be concluded from the above discussion that the attackers always target the popular exchanges to increase their profit. However, it does not implies that individual users are not targeted, it's just that the small attacks go unnoticed. Recently, in August 2016, *BitFinex*, which a popular cryptocurrency exchange suffered a hack due to their wallet vulnerability, and as a result, around 120000 bitcoins were stolen.

From the nature of the aforementioned attacks, it can be concluded that security is a vital concern and biggest weakness for cryptocurrency marketplaces and exchanges. In particular, as the number of bitcoins stored and their value has skyrocketed over the last year, Bitcoin digital wallets have increasingly become a target for hackers. At the social level, what is obvious and does not need mentioning (although some, amazingly, dispute it) is that individuals who handle our bitcoins should be public figures with their full background on display for otherwise they cannot be held accountable. Lacking such accountability, hundreds of millions, understandably, is far too tempting as we have often seen. The equally important point is that bitcoins security is very hard. Exchanges, in particular, require highly experienced developers who are fully familiar with the Bitcoin protocol, the many aspects of exchange coding and how to secure hard digital assets for, to truly secure Bitcoin, exchanges need layers and layers amounting to metaphorical armed guards defending iron gates with vaults deep underground behind a thousand doors. Furthermore, we believe that in future, the attack vector on Bitcoin will not be limited just to stealing the bitcoins, and it will also include threats such as slow down the Bitcoin adoption, reduce the efficiency of Bitcoin infrastructures, and slowdown the Bitcoin development.

IV. Security: Countermeasures for Bitcoin Attacks

In this section, we discuss the state-of-the-art security solutions that provides possible countermeasures for the array of attacks (please refer to Section III) on Bitcoin and its underlying technologies.

A. No more double spending

The transaction propagation and mining processes in Bitcoin provide an inherently high level of protection against double spending. This is achieved by enforcing a simple rule that only unspent outputs from the previous transaction may be used in the input of a next transaction, and the order of transactions is specified by their chronological order in the blockchain which is enforced using robust cryptography techniques. This boils down to a distributed consensus algorithm and time-stamping. In particular, the default solution that provides resistance to double spending in Bitcoin is its use of Proof-of-work (PoW) based consensus algorithm, which limits the capabilities of an adversary concerning, the computational resources available to an adversary and the percentage of honest miners in the network. More specifically, the purpose of the PoW is to reach consensus in the network regarding the blockchain history, thereby synchronizing the transactions or blocks and making the users secure against double-spending attacks. Moreover, the concept of PoW protect the network against being vulnerable to sybil attack because a successful sybil attack 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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

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 Bitcoin address that provides pseudonymity to its users, and the lack of transaction rollback scheme once it is successfully added in the blockchain. Therefore, the latter approach, i.e., prevent double spend, is desirable in Bitcoin.

The most effective yet simple way to prevent a double spend is to wait for multiple numbers of confirmations before delivering goods or services to the payee. In particular, the possibility of a successful double spend decreases with increase in the number of confirmations received. Of course, the longer back transactions lie in the blockchain, the more blocks need to be caught up until a malicious chain gets accepted in the network. This limits attacker from possible revise the history of transactions in the chain. For instance, unconfirmed Bitcoin transaction (zero block transaction) has a high risk of double spend, while a transaction with at least one confirmation has *statically* zero risks of double spend, and a transaction with six confirmations are commonly considered steady, hence has zero risks of double spend. The classic Bitcoin client will show a transaction as *not unconfirmed* until the transaction is six blocks deep[5] in the blockchain. However, waiting of six transactions (about one hour) might not be suitable for various applications such as fast payment systems, e.g., Alice is very hungry and she wants to buy a snack with bitcoins. There is nothing special about the choice of the default *safe* confirmation value, i.e., six confirmations. The choice is based on the assumption that an adversary is unlikely to control more than 10% of the mining power, and that a negligible risk lower than 0.1% is acceptable. It means that on one hand, the six confirmations are overkill for casual attackers, while at the same time it is powerless against more dedicated attackers with much more than 10% mining power.

Authors in [2] evaluate three techniques that can be used to detect possible double spending in fast payment systems. The three techniques are as follow: *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 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 can *see* a number of transactions in the network during its *listening period*, thus it 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 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%). Although the *GHash.IO* remained honest by transferring a part of its mining power to other pools. However, the incentives that motivate an adversary to create large pools remains in the network, always 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 [81]. 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. Hence 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 needs to install mining equipment required to solve Y on time. It is estimated that GHash.IO owns only a small percentage of the network's computing power regarding hardware, as the pool shrank significantly after public outrage. Depending on the difficulty of $Y's$ cryptographic puzzle this would only allow a certain number of untrusted individuals to join. In this way, as GHash.IO is a public pool, severely limit its size.

Authors in [134] propose the use of decentralized non-equivocation contracts, to detect the double spending and penalize the malicious payer. The basic idea of non-equivocation contracts is that the payer locks some bitcoins in a deposit when he initiates a transaction with the payee. If the payer double spends, a cryptographic primitive called accountable assertions can be used to reveal his Bitcoin credentials for the deposit. Thus, the malicious payer could be penalized by the loss of deposit coins. However, such decentralized non-equivocation contracts are subjected to collusion attacks where the payer colludes with the beneficiary of the deposit and transfers the Bitcoin deposit back to himself when he double spends, resulting in no penalties. On the other hand, even if the beneficiary behaves honestly, the victim payee cannot get any compensation directly from the deposit in the original design. To prevent such collusion attacks, authors in [135] design fair deposits for Bitcoin transactions to defend against double-spending. The fair deposits ensure that the payer will be penalized by the loss of his deposit coins if he double spends and the victim payees loss will be compensated. The proposed protocol uses the assertion scheme from [134]. In particular, the beneficiary can recover the payers secret key if the payer double spends. However, to ensure that the payees loss can be compensated if the payer double spends, in addition to a signature generated with the payers secret key, a signature

[5]Each new block that will be put on top of a block containing the desired transaction will result in the generation of a confirmation for the desired transaction.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

created with the payees secret key is required for the release of the compensation locked in the deposit. Meanwhile, the incentive for the beneficiary is also guaranteed in the deposit.

Another solution to control double spending was proposed in [136] 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 it is given to the victim who encountered the loss. Due to the punishing attribute of the network, the attack can be controlled. In [76], 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 this 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 and take primitive actions in future.

Solution for 50% attack is presented in [76]. 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 a delay in the propagation of the legitimately discovered blocks, and the attacker advertises her block selfishly. Hence, the advertiser's blockchain will increase, and the other blocks receive less attention due to delay. 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. While 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 by controlling high hash-power compared with other nodes in the network. The solution to this attack is same as that of block discarding attack. In [137], 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 attack more difficult for adversaries. 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 deals in bitcoins 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 attack, to name a few. These attacks aim to cheat the mining incentive system of Bitcoin. Therefore, at any point in time, detecting and mitigating the faulty forks from the set of available forks poses a significant challenge for Bitcoin protocol developers. The most straightforward solution to handle the selfish mining is suggested in [6]. The authors propose a simple, backward-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 [6], authors in [87] 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 [138] 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 [85]. 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. Hence it will not get benefits in winning the fork race. Authors in [139] proposes another defense against selfish mining, in which miners need to publish intermediate blocks (or in-blocks). These blocks, although are valid with lower puzzle difficulty, but confer no mining reward onto the miner who discovers one. When a fork happens, miners adopt the branch with the largest total amount of work, rather than the longest chain.

Unlike most of the aforementioned solutions against malicious forking, authors in [86] proposes 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 the 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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

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 dummy block called *Zeroblock*. These signed dummy Zeroblocks will accompany the solved blocks to prove, that the block is witnessed by the network and that a competing block is absent before miners can work on it. For forking attacks that are internal to a pool, authors in [84] 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 who 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 [97], 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 the network. 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 usual mining routine. Also, 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, hence makes it difficult for the manager to keep the process of counter-bribing active for more extended periods.

C. Securing Bitcoin wallets

A wallet contains private keys, one for each account [126]. 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 [140]. 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 [123] for secret sharing and threshold cryptography.

A manual method of wallet protection was proposed by [141] called "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 the private key of a user. Authors in [141] claim that if the computer is not connected to the Internet, the hackers will not get to know the keys. Hence the wallet safety can be achieved. Securing wallets with new cryptographic algorithms apart from ECDSA is still an open issue and a challenge. In [142], an article states that US government have launched their 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 [106], authors propose *BlueWallet*, a proof-of-concept based hardware token for the authorization of transactions to protect the private keys. The concept is similar to the use of the "cold wallet", that is, it uses 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 at 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 which provides 2-of-3 multi-signature transactions to its clients. However, the drawback of using the multi-signature transactions is that it significantly compromises the privacy and anonymity of the user. Authors in [104] propose an efficient and optimal threshold Digital Signature Algorithm (DSA) scheme for securing private keys. The main idea behind the use of threshold signatures proposed in [104] is derived from secret sharing [143], in which the private key is split into shares. Any subset of the shares that is equal to or greater than a predefined threshold can reconstruct the private key, but any subset that is smaller will gain no information about the key. The main property of threshold signatures [105] is that the key is never revealed because the participants directly construct a signature. Recently, authors in [107] present a TrustZone[6] based Bitcoin wallet and showed that it is more resilient to the dictionary and side-channel attacks. Although the use of TrustZone makes use of the encrypted storage, hence the writing and reading operations become slower.

D. Securing Bitcoin Networks

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

1) DDoS Attacks: In [112], authors propose a game theoretic approach for analyzing the DDoS attacks. The game assumes that the pools compete 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. In this way, authors draw an equilibrium condition between the players and conclude that the larger pools will have more incentives against the smaller pools. In [9], authors propose a "miner's dilemma", again a game theoretical approach to model the behavior of miners similar to repetitive prisoner's dilemma. There exist a game between the pools. The longest chain dominates over

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

the smaller chains and grabs the rewards by behaving selfishly in the network. Game theory concludes that by performing attacks, the pools 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 [113], authors propose Proof-of-Activity (PoA) protocol, which is robust against a DDoS 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 honest. Any subsequent storage of transactions in 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, then the trustworthiness among other peers increases and more miners get attracted towards the chain. Hence, an adversary cannot place a malicious block or 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 [144], 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 network until debugged. Other possible methods to protect against DDoS attacks include: (i) configure the network in a way that malicious packets and requests from additional 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 applied 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 at 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. Hence it can isolate the victim node. A set of techniques is suggested in [109] to avoid time jacking that includes, use of the system time instead of 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. Furthermore, node timestamps can be made dependent on the blockchain timestamps [109].

In [3], 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 other peers in the network, these peers 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 unique intrusion detection system to check the misbehaving nodes in the network. The addresses which misbehave in the network could be banned from connections. These features can prevent the users from an eclipse attack. 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 [100], 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 with 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 poses a threat to the customer privacy. 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 phase 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 [145], 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. Besides, all the transactions in the Bitcoin network should have confirmations. In [102], authors show a case of malleability attack on "deposit protocol", and provides a solution namely *new deposit protocol*. Finally, the new Segregated Witness [7] (SegWit) proposal stores transaction signatures in a separate Merkle tree, prevent unintended transaction malleability. Moreover, it further enables advanced second-layer protocols such as the Lightning Network, MAST, atomic swaps, and more. Recently, the SegWit soft fork has been activated on the Bitcoin network. More specifically, the SegWit activation means that the block size limit is replaced by a block "weight" limit, which allows for blocks to the size of 2 MB instead of 1 MB.

4) Reducing Delays in Processing and Propagation of Transactions: In practice, the transactions with a large number of bitcoins are not usually carried out due to the risk of losing it or fear of fraudulent activities. Such transactions are broken into a set of smaller transactions. However, this eventually increases the delay in completing the transaction because the network has to validate more number of transactions. Therefore to reduce this delay, authors in [42] suggest performing the payments offline through a separate type of transactions called "micropayments" [146] 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 miners verify it. This happens for all the transactions, and the network might get clogged at places where a large number of transaction exists. Also, in such situations, the network gives preference to transactions with large denomination and transaction fees as compared to the smaller ones. Hence, by establishing micro-

[7] $https://en.bitcoin.it/wiki/Segregated_Witness$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

payment channels, the separate dedicated channel is allocated for the counter-parties 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 transactions that they want to perform. If either of the ones misbehaves, then the transaction is broadcasted for the verification in the Bitcoin network. The channels obey the Bitcoin protocols, and they are established like any other naive network routing techniques. Hence, these micropayment channels constitute a "lightning network" within the Bitcoin network. The advantages of using such 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, counter-parties 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.

Transactions propagation delay in Bitcoin network facilitates the double-spending attack. Hence accelerating the transaction propagation will help to reduce the probability of performing a successful double-spending attack. Authors in [147] propose a Bitcoin Clustering Based Ping Time protocol (BCBPT) to minimize the transaction propagation delay by using the proximity information (e.g., ping latencies) while connecting to peers. Moreover, in the context of the selfish mining attack, authors in [148] study the effect of communication delay on the evolution of the blockchain.

In [58], 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 to 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 to 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 at every sender and pattern can be analyzed. If a transaction is not replied by a node in a time window, then the sender could ask other nodes to confirm the transaction.

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

The security issues in Bitcoin are closely linked with the transaction privacy and user anonymity. The systematic monitoring of the Bitcoin's unencrypted peer-to-peer network and analysis of the public blockchain can reveal a lot of information such as who is using Bitcoin and for what purposes. Additionally, the use of *Know Your Customer* (KYC) policies and *Anti-Money Laundering* (AML) regulation with network traffic and blockchain analysis techniques, could further enhance the quality of the extracted information. From privacy as well as business perspectives, this is not good. For instance, users might not necessarily want the world to know where they spend their bitcoins, how much they own or earn. Similarly, the businesses may not want to leak transaction details to their competitors. Furthermore, the fact that the transaction history of each bitcoin is traceable puts the fungibility of all bitcoins at risk. To this end, we discuss the threats and their existing countermeasures for enabling privacy and enhancing anonymity for Bitcoin users in the following section.

V. Privacy and Anonymity in Bitcoin

The traditional banking system achieves a level of privacy by limiting access to transactions information to the entities involved and the trusted third party. While in Bitcoin, the public blockchain reveals all the transaction data to any user connected to the network. However, the privacy can still be maintained up to a certain level by breaking the flow of information somewhere in the Bitcoin transaction processing chain. Bitcoin achieves it by keeping public keys anonymous, i.e., the public can see that someone is sending an amount to someone else, but without information linking the transaction to anyone. To further enhance the user privacy, it is advised to use a new key pair for each transaction to keep them from being linked to a particular user. However, linking is still possible in multi-input transactions, which necessarily reveal that their inputs were owned by the same owner. Also, if the owner of a key is revealed, there is a risk that linking could reveal other transactions belonging to the same user. In particular, Bitcoin offers a partial unlinkability (i.e., pseudonymity), and thus it is possible to link some transactions to an individual user by tracing the flow of money through a robust blockchain analysis procedure. Bitcoin technology upholds itself when it comes to the privacy, but the only privacy that exists in Bitcoin comes from pseudonymous addresses (public keys or their hashes) which are fragile and easily compromised through different techniques. These technique includes, 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. In [115] authors highlight the fact that the Bitcoin 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. Recently, authors in [64] presents a comprehensive survey which includes an overview and detailed investigation of anonymity and privacy in different cryptocurrency systems. In this section, we discuss the various security threats to privacy and anonymity specific to the Bitcoin users and the corresponding state-of-the-art solutions that are proposed to enhance the same.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

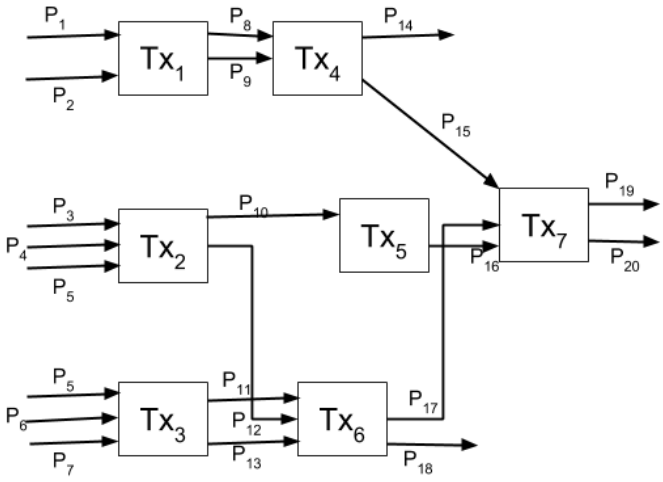


Fig. 10. Blockchain analysis - Transaction graph

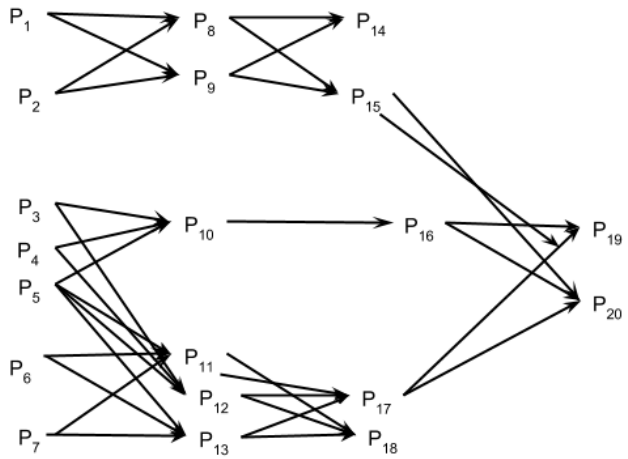


Fig. 11. Blockchain analysis - Address graph

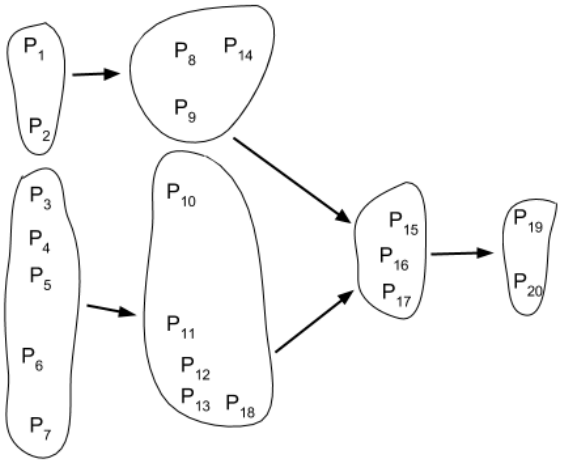


Fig. 12. Blockchain analysis - Entity/User graph

A. Blockchain Analysis and Deanonymization

Complete anonymity in Bitcoin is a complicated issue. To enforce anonymity in transactions, the Bitcoin allows users to generate multiple Bitcoin addresses, and it only stores the mapping information of a user to her Bitcoin addresses on the user's device. As a user can have multiple addresses, hence an adversary who is trying to deanonymize needs to construct a one-to-many mapping between the user and its associated addresses. In particular, the Bitcoin users can be linked to a set of public addresses by using a full blockchain analysis procedure [62]. Authors in [115] show that the two non-trivial networking topologies called *transaction network* and *user network*, which provides reciprocal views of the Bitcoin network and have possible adverse implications for user anonymity. Similar to work done in [115], authors in [149] presents an evaluation for privacy concerns in Bitcoin by analyzing the public blockchain. The analysis of blockchain requires three pre-processing steps, which includes:

- *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 bitcoins between transactions in the blockchain over time. The set of input and output bitcoins 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 bitcoins (C_i) that forms an input for that transaction. Figure 10 shows an instance of transaction graph for a set of transactions stored in the blockchain.
- *Address graph*: By traversing the transaction graph we can easily infer the relationship between various input and output Bitcoin addresses, 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 11 shows an address graph derived from Figure 11.
- *User/entity graph*: By using the address graph along with a number of heuristics which are derived from Bitcoin protocols, the next step is to create an *entity graph* by grouping addresses that seem to be belonging 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 12 shows the entity graph derived from Figure 11 based on a set of heuristics.

In [149], 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 [150]. 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 heuristic also supports

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

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 10, 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 the same 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 11, 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 12. Authors in [150] argued that the heuristics mentioned above 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 the false positives, authors in [150] suggest the manual inspection process to 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 [140] authors exploit the multi-signature addressing technique for adverse effect on 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 [150] 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 [151] [152].

A commercial approach for blockchain analysis could be to use the software BitIodine [153] 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 there 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" models. 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 [154] propose *BitConeView*, a graphical tool for the visual analysis of bitcoins flow in a blockchain. BitConeView allows to graphically track how bitcoins from the given sources (i.e., transaction inputs) are spent over time using transactions and are eventually stored at multiple destinations (i.e., unspent transaction outputs).

Recently, authors in [155] analyze the impact of online tracking on the privacy of Bitcoin users. The paper shows that if a user purchases by paying with cryptocurrency such as Bitcoin, an adversary can uniquely identify the transaction on the blockchain by making use of the third-party trackers which typically possess enough information about the purchase. Latter, these transactions could be linked to the user cookies and then with the real identity of a user, and user's purchase history is revealed. Furthermore, if the tracker can link the two purchases of the same user to the blockchain in this manner, it can identify the user's entire cluster of Bitcoin addresses and transactions on the blockchain through the use of standard tracking software and blockchain analysis techniques. The authors show that these attacks are resilient against the existing blockchain anonymity techniques such as *CoinJoin* [118]. Also, these attacks are passive, hence can be retroactively applied to past purchases as well.

Finally, network de-anonymization could be used to link an IP address to a user in the Bitcoin's P2P network because while broadcasting a transaction the node leaks its 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 network. Over the years, multiple deanonymization attacks in which an adversary uses a "supernode" that connects with the active peers and listen to the transaction traffic relayed by honest nodes in the network [116] [156] [115] 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% [116]. Moreover, the use of "supernode" for linking is trivial. Hence it 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 more sophisticated network traffic analyzing techniques.

B. Proposals for enabling privacy and improving anonymity

Privacy is not defined as an inherent property in Bitcoin initial design, but it is strongly associated with the system. Therefore, in recent years an array of academic research [149] [165] [166] [153] which shows various privacy-related weaknesses in the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

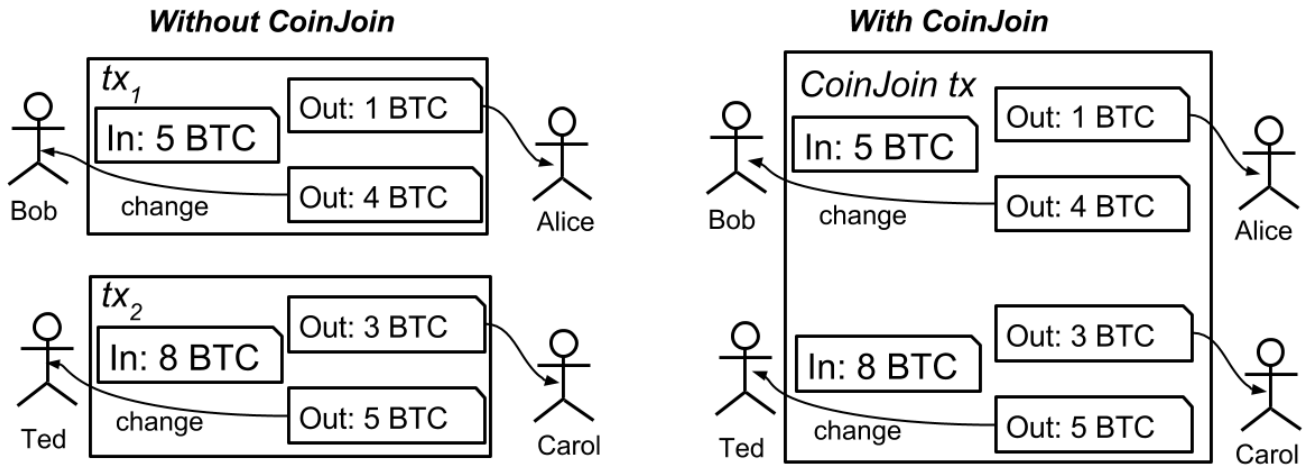


Fig. 13. Example: CoinJoin basic idea

current Bitcoin protocol(s) has been surfaced. This research triggered a large set of privacy-enhancing technologies [165] [167] [114] [162] [164] [168] [157] [161] aiming at strengthening privacy and improving anonymity without breaking Bitcoin 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.

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 tables IV and V) for enabling privacy in cryptocurrencies can be broadly classified into three major categories namely, *Peer-to-peer mixing protocols*, *Distributed mixing networks*, and *Altcoins*.

1) Peer-to-peer mixing protocols: Mixers are anonymous service providers that uses mixing protocols to confuse the trails of transactions. In mixing process, the client's funds are divided into smaller parts. These parts are mixed at random with similar random parts of other clients, and you end up with entirely new coins. This helps to break any link between the user and the coins she purchased. However, mixers are not an integral part of Bitcoin, but various mixing services are heavily used to enhance the anonymity and unlinkability in the system. In peer-to-peer (P2P) mixing protocols [169] [160] [119], 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 participants by permuting ownership of their coins. The goal is to prevent an attacker who 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 depends on the number of users in the anonymity set.

Table IV shows a range of P2P mixing protocols along with their brief description, advantages, and disadvantages concerning user anonymity and transaction security. CoinJoin [118], a straightforward protocol for implementing P2P mixing, it aims to enhance privacy and securely prevent thefts. Figure 13 shows CoinJoin basic idea with an example in which two transactions (i.e., tx_1 and tx_2) are joined into one while inputs and outputs are unchanged. 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 a transaction if its desired output appears in the output addresses of the transaction. In this way, CoinJoin makes the multiple inputs of a transaction independent from each other. Thus it breaks the fundamental heuristic from Section V-A (i.e., inputs of a transaction belong to the same user). However, CoinJoin has few major drawbacks, which includes 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 internal unlinkability issue and to increase the robustness to DoS attacks, authors in [119] propose CoinShuffle, a decentralized protocol that coordinates CoinJoin transactions using a cryptographic mixing technique. Later, an array of protocols [157] [158] [160] are built on the concept of either CoinJoin or CoinShuffle that enhances the P2P mixing by providing various improvements, that includes resistance to DoS, sybil, and intersection attacks, plausible deniability, low mixing time, and scalability of the mixing groups.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

TABLE IV. TECHNIQUES FOR IMPROVING PRIVACY AND ANONYMITY IN BITCOIN

Proposals	Type/Class	Distinct features and properties	Advantages	Disadvantages
CoinJoin [118]	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 (by stalling joint transactions), sybil and intersection attacks, prevents plausible deniability
CoinShuffle [119]	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 [114]	P2P	anonymously partnering and multi-round mixing	distributed pairing, internal unlinkability, thwarts sybil and DoS attacks	higher mixing time
CoinShuffle++ / *DiceMix* [157]	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 [158]	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 [159]	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 [160]	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 [161]	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 [162]	Distributed	third-party mixing with accountability	DoS and sybil resistance	partial internal unlinkability and theft resistance,
BlindCoin [163]	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 [164]	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

2) Distributed mixing networks: Authors in [162] 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 [163] 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 additional transactions to publish and redeem the blinded tokens, and the threat of theft from the mix is still present.

Recently, in [164] authors propose *TumbleBit*, a Bitcoin-compatible unidirectional unlinkable payment hub that allows peers to make fast, off-blockchain payments anonymously through an untrusted intermediary called *Tumbler*. Similar to Chaumian original eCash protocol [170], TumbleBit enforces anonymity in the mixing by ensuring that no one, not even the Tumbler, can link a transaction of its sender to its receiver. The mixing of payments from 800 users shows that TumbleBit provides strong anonymity and theft resistance and it is scalable.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

TABLE V. SUMMARY OF ALTCOINS

Proposals	Distinct features and properties	Advantages	Disadvantages
ZeroCoin / ZeroCash / Zcash [171] [167]	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 [172]	relies on ring signatures to provide anonymity	provides strong privacy and anonymity guarantees	higher computational complexity, not compatible with pruning
MimbleWimble [173] [174]	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 [175]	Bitcoin-like cryptocurrency with strong consistency via collective signing	lower consensus latency and high transaction throughput, resistance to selfish and stubborn mining [8], eclipse and delivery-tampering and double-spending attacks	vulnerable to slow down or temporary DoS attack and 51% attack,
Ethereum (ETH) [176]	uses proof-of-stake, open-ended decentralized software platform that enables Smart Contracts and Distributed Applications	run without any downtime, fraud, control or interference from a third party, support developers to build and publish distributed applications	scalability issues (uses complex network), running untrusted code, limited (i.e., non-turing-complete) scripting language
Mastercoin (or Omni) [177]	uses enhanced Bitcoin Core ad Proof of Authenticity, Colored coins, Exodus address	Easy to use, secure web wallets available, Escrow fund (insurance against panic), Duress protection using a trusted entity	wallets handling the transactions should aware of the concept of colored coins, possibility to accidentally uncolor colored coin assets exists
Litecoin (LTC, litecoin.org)	uses Segwit, which allows technologies like Lightning Network	scalable, low transaction mining time, anonymous and cheaper	very few stores accept payment in Litecoins, high power consumption
Dash (DASH, dashpay.io)	uses Proof of Service, implements native CoinJoin like transactions	higher privacy (mixes transactions using master nodes), InstantX provides faster transaction processing	less liquid,technology is too young, does not yet have a critical mass of merchants or users
Ripple (XRP, ripple.com)	implements a novel low-latency consensus algorithm based on byzantine agreement protocol	fast transaction validation, less energy-intensive, no 51% attack	not fully decentralized, vulnerable to attacks such as consensus split, transaction flood and software backdoor
Monero (XMR, getmonero.org)	based on the CryptoNote protocol,	improves user privacy by using ring signatures, lower transaction processing time (average every 2 minutes)	transaction linkability could be achieved by leveraging the ring signature size of zero, output merging, temporal analysis
Counterparty (XCP, counterparty.io)	created and distributed by destroying bitcoins in a process known as *proof of burn*	same as bitcoins	same as bitcoins

3) Bitcoin extensions or Altcoins: Bitcoin has not just been a most popular cryptocurrency in today's market, but it ushers a wave of other cryptocurrencies that are built on decentralized peer-to-peer networks. In fact, the Bitcoin has become the de facto standard for the other cryptocurrencies. The other currencies which are inspired by Bitcoin are collectively known as *altcoins*. Instead of proposing techniques (such as mixing and shuffling) to increase transaction anonymity and user privacy, the altcoins work as an extension to Bitcoin or a full-fledged currency. The popular altcoins along with their brief description have been shown in Table V. Some of these currencies are easier to mine than Bitcoin. However, there are tradeoffs, including higher risk brought on by lesser liquidity, acceptance, and value retention.

Authors in [171] propose *ZeroCoin*, a cryptographic extension to Bitcoin which provides anonymity by design by applying zero-knowledge proofs which allow fully encrypted transactions to be confirmed as valid. It is believed that this new property could enable entirely new classes of blockchain applications to be built. In ZeroCoin, a user can just wash the linkability traces from its coins by exchanging them for an equal value of ZeroCoins. But unlike the mixing mentioned above 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. For instance, $Alice$ can prove to others that she owns a bitcoin and is thus eligible to spend any other bitcoin. For this purpose, first, she produces a secure commitment, i.e., the zerocoin, which is recorded in the blockchain so that others can validate it. To spend a bitcoin, she broadcasts a zero-knowledge proof for the respective zerocoin, together with a transaction. The zero-knowledge cryptography protects $Alice$ from linking the zerocoin to her. Still, the other

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

participants can verify the transaction and the proof. Instead of a linked list of Bitcoin transactions, Zerocoin introduces intermediate steps. In this way, the use of zero-knowledge proofs prevent the transaction graph analyses. Unfortunately, even though Zerocoins properties may seem appealing, it is computationally complex, bloats the blockchain and requires protocol modifications. However, it demonstrates an alternative, privacy-aware approach. Currently, ZeroCoin derives both its anonymity and security against counterfeiting from strong cryptographic assumptions at the cost of substantially increased computational complexity and size.

An extension of ZeroCoin called *ZeroCash* (also know as Zcash) is presented by [167]. ZeroCash uses an improved version of zero-knowledge proof (concerning functionality and efficiency) called *zk-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 [173] 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.

Beyond Bitcoin, the so-called second generation of cryptocurrencies, such as Ethereum (Ether), Mastercoin (MSC), Counterparty (XCP) are introduced in the market. These cryptocurrencies implement a new transaction syntax with a fully-fledged scripting language written in Turing complete language. Furthermore, these cryptocurrencies understood the digital assets regarding smart contracts and colored coins. Unlike Bitcoin, Ethereum was designed to be much more than a payment system. In particular, it is a decentralized platform that runs smart contracts, which are the applications that run precisely as programmed without any possibility of downtime, censorship, fraud or third-party interference. This implies that these digital assets can be used to realize sophisticated financial instruments such as stocks with automatic dividend payouts or to manage and trade physical properties such as a house. Most of these next-generation coins work on top of Bitcoin blockchain and are therefore also known as *on-chain currencies*. Since they encode their transactions into Bitcoin transactions, they lack the validation of transactions by miners, because Bitcoin miners do not *understand* the new transaction types. For this purpose, a new protocol layer is built upon Bitcoins strong foundation and its security. Furthermore, it is seen as an increase in Bitcoins value from which both will profit. As a detailed discussion on the altcoins is out of the scope of our work, we direct interested readers to the existing literature such as [20] and [178].

As a summary, in this section, the Bitcoin privacy and anonymity concerns have been discussed. It is observed that Bitcoin is pseudo-anonymous as the account is tied to the random and multiple Bitcoin addresses and not to the individual users. With the rapidly increasing popularity of Bitcoin, the need for privacy and anonymity protection also increases, and it must be ensured that the users will receive a satisfactory level of service concerning privacy, security, and anonymity.

VI. Summary of observations and future research directions

After our comprehensive survey on the security and privacy aspects of Bitcoin and its primary related techniques, we now summarize our lessons learned, before presenting the possible future challenges and research directions. Some of these are already discussed in previous sections. However, remaining challenges and open research issues are dealt in brief in this section.

With the use of proof-of-work based consensus algorithm and a secure timestamping service, Bitcoin provides a practical solution to the Byzantine Generals problem. However, to achieve distributed consensus, Bitcoin exposes itself to many security threats as mentioned in Section III. In particular, the primary threat that cannot be eliminated entirely is race attacks for double-spending. The transparency in the system is provided by using an unforgeable distributed ledger (i.e., blockchain), which holds all the transactions ever processed in such a way that anyone can verify their integrity and authenticity. But, at the same time, this transparency introduces a ubiquitous global attacker model. Hence, we can deduce from the discussion presented in Section V that Bitcoin is anything but private. Nevertheless, Bitcoin provides pseudonymity by hiding identities, and the research community is putting a lot of efforts to further strengthen this property. For instance, use of commitment schemes such as zero-knowledge proofs dramatically improves unlinkability and traceability in transactions.

One of the significant contributions of Bitcoin is the degree of transparency and decentralization, which it provides along with the adequate level of security and privacy, which was previously deemed impossible. The original concept of mining, which could be based on proof of work, proof of stake, proof of burns or some other scheme, not only secures the blockchain but it eventually achieves the distributed consensus. In particular, the most important steps that make the whole process so cohesive includes, the way these schemes binds the *votes* to something *valuable*, give rewards in exchange to *pay* for these valuables, and at the same time controls the supply of the cryptocurrencies in the system. Without these mining schemes, the fake identities would be able to easily disturb (through sybil attack) the consensus process and destroy the system. Due to this, i.e., availability of a mining based consensus protocol, we can safely conclude that 51% attacks are the worst case scenario for Bitcoin. However, the rapidly increasing mining pools threatens the decentralization of Bitcoin.

It is hard to comment on the survivability of the Bitcoin, i.e., whether Bitcoin can and will stay as robust as it is today. In particular, the scalability of the network, the continuously decreasing rewards, increasing transaction fee, and the security and privacy threats are the pressing issues, which needs to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

be addressed. The peer-to-peer network already seems to be having the symptoms of degradation, which can be seen in terms of propagation delay for both, the new transaction generated by a user and the newly validated block by a miner. The network propagation delay becomes a significant issue because the Bitcoin security assumptions heavily rely on the fast propagation of transactions and blocks. Therefore, it is very important that the Bitcoin network is easy to scale to more participants and it can handle higher transaction rates. In case of the subsiding mining rewards, the research community is unsure whether this poses a real problem or if fees can provide the necessary incentive. So far, various improvements and altcoins have been implemented to resolve the issues mentioned above (please refer to tables IV and V. However, it remains unclear which of the alternative approaches are most promising in terms of practical implementation that will improve Bitcoin.

From the improvement perspective, Bitcoin can consider all the altcoins as a huge testing environment, from which it can borrow novel techniques and functionalities to address its weaknesses. At least in the recent future, the Bitcoin will constantly be evolving and will be in the underdevelopment phase. Hence we now present few research directions that could be exploited in this direction.

- *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. Such 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, thus a game theoretic approach is well suited for achieving Nash equilibrium among miners (i.e., players) [179]. Attackers may try to contribute to an increase of their chain length compared to honest chain in the network. It 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 [179] [180] [181] which shows that the use of the game-theoretic approaches provides useful information about the effects of selfish mining, block withholding, discarding attacks, and the incentive distribution problem in the mining pools. Therefore, we believe that this approach could be used efficiently for modeling 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 lightweight protocol used for the verification of the transaction sent from a user [182]. It is often vulnerable to attacks such as sybil and double spending. Hence, a more robust verification protocol is a current requirement. For the key manipulations and calculations, a distributed approach is always preferred over the centralized one. This is to avoid the point of failure else the central server is 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 to solve various attacks. For instance, fix a group head and get an additional signature or authentication on every transaction [165]. Another approach is to use "trusted paths" which is based on hardware that allows users to read and write few cryptographic data [165]. Finally, there are few techniques which use the Bloom filters for securing wallets. Nevertheless, filters might lead to false positives (FP) and false negatives (FN) that will consume the network bandwidth, hence reducing the FP and FN could be a potential research directive.
- *Improving blockchain protocol:* For the first time, the blockchain provides a probabilistic solution to the Byzantine Generals problem [183], 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 [184] of venture capital money to exploit this technology for applications such as voting, record keeping, contracts, to name a few. Despite its potential, blockchain protocols face significant concerns regarding their privacy [185] and scalability [47] [186]. The append-only nature of blockchain is essential to the security of the Bitcoin ecosystem because the transactions will be stored in the ledger forever and it is immutable. However, an immutable ledger is not appropriate for all new applications that are being envisaged for the blockchain. Recently, authors in [187] 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 PoW 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 [188], 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 [175] is a future requirement. Recently authors in [189] 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 Bitcoin or any other digital currency to retain its popularity in the market. For instance, the Bitcoin miners may migrate by looking at the rewards given by the other competitors or by the fact that for every 210,000 blocks the incentives are halved. Therefore, essential questions that need to be addressed include, 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 new, but the advent of blockchain technology spurred interest in it. This is because the blockchain eliminates the need to rely on a trusted third party to "execute" the contract, and it enables the use of cryptocurrency as "programmable money". Bitcoin support for smart contracts is insufficient. Recently authors in [190] propose *Hawk*, which uses a blockchain model of cryptography to generate privacy-preserving smart contracts. Similar to Bitcoin, authors in [191] proposes *Enigma*, a decentralized computation platform which provides a highly optimized version of secure multi-party computation with guaranteed privacy to efficiently execute smart contracts.

VII. CONCLUSIONS

Bitcoin has already manifested 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 approximately 1146 different cryptocurrencies in action, out of which many are a recent introduction to the market. From all these fiat-currencies, the outstanding popularity and high market capital of Bitcoin make it attractive for adversaries to launch various security threats. According to our survey, even though the construction of the Bitcoin system with proof-of-work and consensus algorithm to protect the user actions are the robust features in Bitcoin, these itself becoming a point of manipulation for cyber criminals. Starting from packet sniffing to the double spending, the Bitcoin is dreaded with various attacks. Though literature provides solutions against few of these attacks, the robust and practical security solutions that can ensure proper functioning of Bitcoin in the future are still absent. Together with security, the distributed nature of Bitcoin blockchain 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 essential 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 study and emphasize the issues 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 system.

REFERENCES

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

[2] 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.

[3] 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.

[4] C. Decker and R. Wattenhofer, "Bitcoin transaction malleability and mtgox," in *ESORICS 2014: 19th European Symposium on Research in Computer Security*. Springer International Publishing, 2014, pp. 313–326.

[5] A. Maria, Z. Aviv, and V. Laurent, "Hijacking bitcoin: Routing attacks on cryptocurrencies," in *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE, 2017.

[6] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *Financial Cryptography and Data Security: 18th International Conference*. Springer Berlin Heidelberg, 2014, pp. 436–454.

[7] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Financial Cryptography and Data Security: 20th International Conference, FC 2016, Christ Church, Barbados, February 22–26, 2016*. Springer Berlin Heidelberg, 2017, pp. 515–532.

[8] 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.

[9] I. Eyal, "The miner's dilemma," in *Proceedings of the 2015 IEEE Symposium on Security and Privacy*, ser. SP '15. Washington, DC, USA: IEEE Computer Society, 2015, pp. 89–103.

[10] 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.

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

[12] 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.

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

[14] 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.

[15] 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.

[16] S. Huckle, R. Bhattacharya, M. White, and N. Beloff, "Internet of things, blockchain and shared economy applications," *Procedia Comput. Sci.*, vol. 98, pp. 461–466, Oct. 2016.

[17] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. A. Ogah, and Z. Sun, "Blockchain-based dynamic key management for heterogeneous intelligent transportation systems," *IEEE Internet of Things Journal*, no. 99, 2017.

[18] M. Mettler, "Blockchain technology in healthcare: The revolution starts here," in *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*, Sept 2016, pp. 1–3.

[19] 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.

[20] 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.

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

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

[23] 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.

[24] A. Back, "Hashcash - a denial of service counter-measure," *Available: http://www.hashcash.org/papers/hashcash.pdf*, 2002.

[25] 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.

[26] K. Kaskaloglu, "Near zero bitcoin transaction fees cannot last forever," 2014, pp. 91–99.

[27] D. Easley, M. O'Hara, and S. Basu, "From mining to markets: The evolution of bitcoin transaction fees," *Available: http://dx.doi.org/10.2139/ssrn.3055380*, 2017.

[28] M. Möser and R. Böhme, "Trends, tips, tolls: A longitudinal study of bitcoin transaction fees," pp. 19–33, 2015.

[29] M. Carlsten, H. Kalodner, S. M. Weinberg, and A. Narayanan, "On the instability of bitcoin without the block reward," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. ACM, 2016, pp. 154–167.

[30] R. C. Merkle, "A digital signature based on a conventional encryption function," in *Advances in Cryptology — CRYPTO '87: Proceedings*. Springer Berlin Heidelberg, 1988, pp. 369–378.

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

[32] A. Laszka, B. Johnson, and J. Grossklags, "When bitcoin mining pools run dry," in *Financial Cryptography and Data Security: FC 2015 International Workshops, BITCOIN, WAHC, and Wearable, San Juan, Puerto Rico, January 30, 2015*. Springer Berlin Heidelberg, 2015, pp. 63–77.

[33] O. Schrijvers, J. Bonneau, D. Boneh, and T. Roughgarden, *Incentive Compatibility of Bitcoin Mining Pool Reward Functions*. Springer Berlin Heidelberg, 2017, pp. 477–498.

[34] M. Peck, "Adam back says the bitcoin fork is a coup," *Available: http://spectrum.ieee.org/tech-talk/computing/networks/the-bitcoin-for-is-a-coup*, 2015.

[35] "How a visa transaction works," *Available: http://web.archive.org/web/20160121231718/http://apps.usa.visa.com/merchants/become-a-merchant/how-a-visa-transaction-works.jsp*, 2015.

[36] "Paypal: total payment volume 2014-2017," *Available: https:https://www.statista.com/statistics/277841/paypals-total-payment-volume/*, 2017.

[37] J. Gbel and A. E. Krzesinski, "Increased block size and bitcoin blockchain dynamics," in *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*, Nov 2017, pp. 1–6.

[38] P. Wuille, "Segregated witness and its impact on scalability," *SF Bitcoin Devs Seminar*, 2015.

[39] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," 2016.

[40] C. Burchert, C. Decker, and R. Wattenhofer, "Scalable funding of bitcoin micropayment channel networks," in *Stabilization, Safety, and Security of Distributed Systems*, 2017.

[41] J. Hilliard, "Reduced threshold segwit masf," *Available: https://github.com/bitcoin/bips/wiki/Comments:BIP-0091 [Online; accessed 13-November- 2017]*, 2017.

[42] C. Decker and R. Wattenhofer, "A fast and scalable payment network with bitcoin duplex micropayment channels," in *Stabilization, Safety, and Security of Distributed Systems: 17th International Symposium, SSS 2015*. Springer International Publishing, 2015.

[43] P. McCorry, M. Möser, S. F. Shahandasti, and F. Hao, "Towards bitcoin payment networks," in *Information Security and Privacy*. Springer, 2016.

[44] G. Malavolta, P. Moreno-Sanchez, A. Kate, M. Maffei, and S. Ravi, "Concurrency and privacy with payment-channel networks," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. ACM, 2017.

[45] J. Herrera-Joancomartí and C. Pérez-Solà, "Privacy in bitcoin transactions: New challenges from blockchain scalability solutions," in *Modeling Decisions for Artificial Intelligence*. Springer, 2016.

[46] E. Heilman, F. Baldimtsi, and S. Goldberg, "Blindly signed contracts: Anonymous on-blockchain and off-blockchain bitcoin transactions," in *Financial Cryptography and Data Security*. Springer, 2016.

[47] Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in bitcoin," in *Financial Cryptography and Data Security: 19th International Conference*. Springer Berlin Heidelberg, 2015, pp. 507–527.

[48] 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.

[49] S. King and S. Nadal, "Ppcoin: peer-to-peer crypto-currency with proof-of-stake," *Available: http://peercoin.net/assets/paper/peercoin-paper.pdf*, Tech. Rep., Aug. 2012.

[50] A. Miller, A. Juels, E. Shi, B. Parno, and J. Katz, "Permacoin: Repurposing bitcoin work for data preservation," in *2014 IEEE Symposium on Security and Privacy*, May 2014, pp. 475–490.

[51] B. Sengupta, S. Bag, S. Ruj, and K. Sakurai, "Retricoin: Bitcoin based on compact proofs of retrievability," in *Proceedings of the 17th International Conference on Distributed Computing and Networking*, ser. ICDCN '16. ACM, 2016, pp. 14:1–14:10.

[52] I. Bentov, A. Gabizon, and A. Mizrahi, *Cryptocurrencies Without Proof of Work*. Springer Berlin Heidelberg, 2016, pp. 142–157.

[53] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol with chains of variable difficulty," in *Advances in Cryptology – CRYPTO 2017: 37th Annual International Cryptology Conference, Santa Barbara, CA, USA*. Springer International Publishing, 2017, pp. 291–323.

[54] 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.

[55] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, and G. Danezis, "Consensus in the age of blockchains," *CoRR*, vol. abs/1711.03936, 2017.

[56] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys Tutorials*, vol. 7, no. 2, pp. 72–93, 2005.

[57] A. Miller and R. Jansen, "Shadow-bitcoin: Scalable simulation via direct execution of multi-threaded applications," in *8th Workshop on*

Cyber Security Experimentation and Test (CSET 15). Washington, D.C.: USENIX Association, 2015. [Online]. Available: https://www.usenix.org/conference/cset15/workshop-program/presentation/miller

[58] 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.

[59] N. T. Courtois, P. Emirdag, and D. A. Nagy, "Could bitcoin transactions be 100x faster?" in *2014 11th International Conference on Security and Cryptography (SECRYPT)*, Aug 2014, pp. 1–6.

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

[61] A. Gervais, G. O. Karame, V. Capkun, and S. Capkun, "Is bitcoin a decentralized currency?" *IEEE Security Privacy*, vol. 12, no. 3, pp. 54–60, May 2014.

[62] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *Financial Cryptography and Data Security: 17th International Conference, FC 2013*. Springer Berlin Heidelberg, 2013, pp. 6–24.

[63] D. D. F. Maesa, A. Marino, and L. Ricci, "Uncovering the bitcoin blockchain: An analysis of the full users graph," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2016, pp. 537–546.

[64] M. C. K. Khalilov and A. Levi, "A survey on anonymity and privacy in bitcoin-like digital cash systems," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2018.

[65] Q. Wang, X. Li, and Y. Yu, "Anonymity for bitcoin from secure escrow address," *IEEE Access*, vol. 6, pp. 12 336–12 341, 2018.

[66] Y. Liu, X. Liu, C. Tang, J. Wang, and L. Zhang, "Unlinkable coin mixing scheme for transaction privacy enhancement of bitcoin," *IEEE Access*, 2018.

[67] P. Fairley, "Blockchain world - feeding the blockchain beast if bitcoin ever does go mainstream, the electricity needed to sustain it will be enormous," *IEEE Spectrum*, vol. 54, no. 10, pp. 36–59, October 2017.

[68] K. J. O'Dwyer and D. Malone, "Bitcoin mining and its energy footprint," in *25th IET Irish Signals Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014)*, 2014, pp. 280–285.

[69] C. Domingo, "The bitcoin vs visa electricity consumption fallacy," *Available: https://hackernoon.com/the-bitcoin-vs-visa-electricity-consumption-fallacy-8cf194987a50*, 2017.

[70] K. Liao, Z. Zhao, A. Doupe, and G. J. Ahn, "Behind closed doors: measurement and analysis of cryptolocker ransoms in bitcoin," in *2016 APWG Symposium on Electronic Crime Research (eCrime)*, June 2016, pp. 1–13.

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

[72] 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.

[73] K. Krombholz, A. Judmayer, M. Gusenbauer, and E. Weippl, "The other side of the coin: User experiences with bitcoin security and privacy," in *Financial Cryptography and Data Security: 20th International Conference, FC 2016, Christ Church, Barbados*. Springer Berlin Heidelberg, 2017, pp. 555–580.

[74] 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.

[75] 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.

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

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

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

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

[80] 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-pools/*, 2014.

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

[82] 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.

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

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

[85] R. Zhang and B. Preneel, "Publish or perish: A backward-compatible defense against selfish mining in bitcoin," in *Topics in Cryptology – CT-RSA 2017: The Cryptographers' Track at the RSA Conference 2017, San Francisco, CA, USA, February 14–17, 2017, Proceedings*. Springer International Publishing, 2017, pp. 277–292.

[86] S. Solat and M. Potop-Butucaru, "Brief announcement: Zeroblock: Timestamp-free prevention of block-withholding attack in bitcoin," in *Stabilization, Safety, and Security of Distributed Systems: 19th International Symposium*. Springer International Publishing, 2017, pp. 356–360.

[87] E. Heilman, "One weird trick to stop selfish miners: Fresh bitcoins, a solution for the honest miner," 2014.

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

[89] 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.

[90] Y. Kwon, D. Kim, Y. Son, E. Vasserman, and Y. Kim, "Be selfish and avoid dilemmas: Fork after withholding (faw) attacks on bitcoin," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. ACM, 2017, pp. 195–209.

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

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

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

[94] 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.

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

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

[97] 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.

[98] 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.

[99] 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.

[100] P. McCorry, S. F. Shahandashti, and F. Hao, "Refund attacks on bitcoin's payment protocol," in *Financial Cryptography and Data Security: 20th International Conference*. Springer Berlin Heidelberg, 2017, pp. 581–599.

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

[102] M. Andrychowicz, S. Dziembowski, D. Malinowski, and Ł. Mazurek, "On the malleability of bitcoin transactions," in *Financial Cryptography and Data Security: FC 2015 International Workshops, BITCOIN, WAHC, and Wearable*. Springer Berlin Heidelberg, 2015, pp. 1–18.

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

[104] R. Gennaro, S. Goldfeder, and A. Narayanan, "Threshold-optimal dsa/ecdsa signatures and an application to bitcoin wallet security," in *Applied Cryptography and Network Security: 14th International Conference, ACNS 2016*. Springer International Publishing, 2016, pp. 156–174.

[105] 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.

[106] T. Bamert, C. Decker, R. Wattenhofer, and S. Welten, "Bluewallet: The secure bitcoin wallet," *Security and Trust Management: 10th International Workshop, STM 2014*, pp. 65–80, 2014.

[107] 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.

[108] S. Jarecki, A. Kiayias, H. Krawczyk, and J. Xu, "Highly-efficient and composable password-protected secret sharing (or: How to protect your bitcoin wallet online)," in *2016 IEEE European Symposium on Security and Privacy (EuroS P)*, March 2016, pp. 276–291.

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

[110] 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.

[111] M. Vasek, M. Thornton, and T. Moore, "Empirical analysis of denial-of-service attacks in the bitcoin ecosystem," in *Financial Cryptography and Data Security: FC 2014 Workshops, BITCOIN and WAHC 2014,*. Springer Berlin Heidelberg, 2014, pp. 57–71.

[112] B. Johnson, A. Laszka, J. Grossklags, M. Vasek, and T. Moore, "Game-theoretic analysis of ddos attacks against bitcoin mining pools," in *Financial Cryptography and Data Security: FC 2014 Workshops, BITCOIN and WAHC 2014,*. Springer Berlin Heidelberg, 2014, pp. 72–86.

[113] 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.

[114] 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.

[115] P. Koshy, D. Koshy, and P. McDaniel, "An analysis of anonymity in bitcoin using p2p network traffic," in *Financial Cryptography and Data Security: 18th International Conference, FC 2014,*. Springer Berlin Heidelberg, 2014, pp. 469–485.

[116] 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.

[117] 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.

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

[119] T. Ruffing, P. Moreno-Sanchez, and A. Kate, "Coinshuffle: Practical decentralized coin mixing for bitcoin," in *ESORICS 2014: 19th European Symposium on Research in Computer Security*. Springer International Publishing, 2014, pp. 345–364.

[120] 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.

[121] 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.

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

[123] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow, "Elliptic curve cryptography in practice," in *18th International Conference, FC 2014,*. Springer Berlin Heidelberg, 2014, pp. 157–175.

[124] I. Giechaskiel, C. Cremers, and K. B. Rasmussen, "On bitcoin security in the presence of broken cryptographic primitives," in *Computer Security – ESORICS 2016: 21st European Symposium on Research in Computer Security, Heraklion, Greece*. Springer International Publishing, 2016, pp. 201–222.

[125] J. J. Hoch and A. Shamir, "On the strength of the concatenated hash combiner when all the hash functions are weak," in *Automata, Languages and Programming: 35th International Colloquium, ICALP 2008, Reykjavik, Iceland*. Springer Berlin Heidelberg, 2008, pp. 616–630.

[126] S. Eskandari, D. Barrera, E. Stobert, and J. Clark, "A First Look at the Usability of Bitcoin Key Management," *Proceedings of the NDSS Workshop on Usable Security (USEC)*, 2015.

[127] P. Litke and J. Stewart, "Cryptocurrency-stealing malware landscape," 2014.

[128] T. Moore and N. Christin, "Beware the middleman: Empirical analysis of bitcoin-exchange risk," in *Financial Cryptography and Data Security: 17th International Conference*. Springer Berlin Heidelberg, 2013, pp. 25–33.

[129] 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.

[130] 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.

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

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

[133] M. Conti, P. Kaliyar, and C. Lal, "Remi: A reliable and secure multicast routing protocol for iot networks," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ser. ARES '17, 2017, pp. 84:1–84:8.

[134] T. Ruffing, A. Kate, and D. Schröder, "Liar, liar, coins on fire!: Penalizing equivocation by loss of bitcoins," in *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15. ACM, 2015, pp. 219–230.

[135] X. Yu, M. T. Shiwen, Y. Li, and R. D. Huijie, "Fair deposits against double-spending for bitcoin transactions," in *2017 IEEE Conference on Dependable and Secure Computing*, Aug 2017, pp. 44–51.

[136] 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. ACM, 2012, pp. 906–917.

[137] "Solution to sybil attacks and 51*Available: https://letstalkbitcoin.com/blog/post/solution-to-sybil-attacks-and-51-attacks-in-decentralized-networks*, 2014.

[138] M. O. Rabin, "Transaction protection by beacons," vol. 27, no. 2, 1983, pp. 256 – 267.

[139] R. Zhang, "Broadcasting intermediate blocks as a defense mechanism against selfish-mine in bitcoin," *IACR Cryptology ePrint Archive*, vol. 2015, p. 518, 2015.

[140] 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.

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

[142] "Biometric tech secures bitcoin wallet," no. 6, 2015.

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

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

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

[146] B. Rosenberg, "Micropayment systems. handbook of financial cryptography and security," 2010.

[147] M. F. sallal, G. Owenson, and M. Adda, "Proximity awareness approach to enhance propagation delay on the bitcoin peer-to-peer network," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, June 2017, pp. 2411–2416.

[148] J. Gobel, H. Keeler, A. Krzesinski, and P. Taylor, "Bitcoin blockchain-dynamics: The selfish-mine strategy in the presence of propagation delay," vol. 104, no. Supplement C, 2016, pp. 23 – 41.

[149] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun, "Evaluating user privacy in bitcoin," in *Financial Cryptography and Data Security: 17th International Conference, FC 2013*. Springer Berlin Heidelberg, 2013, pp. 34–51.

[150] 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.

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

[152] F. Reid and M. Harrigan, "An analysis of anonymity in the bitcoin system." Springer New York, 2013, pp. 197–223.

[153] M. Spagnuolo, F. Maggi, and S. Zanero, "Bitiodine: Extracting intelligence from the bitcoin network," in *Financial Cryptography and Data Security: 18th International Conference, FC 2014*. Springer Berlin Heidelberg, 2014, pp. 457–468.

[154] 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.

[155] S. Goldfeder, H. A. Kalodner, D. Reisman, and A. Narayanan, "When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies," *CoRR*, 2017.

[156] 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.

[157] T. Ruffing, P. Moreno-Sanchez, and A. Kate, "P2p mixing and unlinkable bitcoin transactions," 2017.

[158] T. Ruffing and P. Moreno-Sanchez, "Valueshuffle: Mixing confidential transactions for comprehensive transaction privacy inbitcoin," in *Financial Cryptography and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017*. Springer International Publishing, 2017, pp. 133–154.

[159] S. Bojja Venkatakrishnan, G. Fanti, and P. Viswanath, "Dandelion: Redesigning the bitcoin network for anonymity," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 1, pp. 22:1–22:34, Jun. 2017.

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

[161] 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.

[162] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, "Mixcoin: Anonymity for bitcoin with accountable mixes," in *18th International Conference, FC 2014*. Springer Berlin Heidelberg, 2014, pp. 486–504.

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

[164] 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.

[165] S. Barber, X. Boyen, E. Shi, and E. Uzun, "Bitter to better — how to make bitcoin a better currency," in *Financial Cryptography and Data Security: 16th International Conference, FC 2012*. Springer Berlin Heidelberg, 2012, pp. 399–414.

[166] S. Meiklejohn and C. Orlandi, "Privacy-enhancing overlays in bitcoin," in *Financial Cryptography and Data Security: FC 2015 International Workshops, BITCOIN, WAHC, and Wearable, San Juan, Puerto Rico, January 30, 2015*. Springer Berlin Heidelberg, 2015, pp. 127–141.

[167] 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.

[168] E. Heilman, F. Baldimtsi, and S. Goldberg, "Blindly signed contracts: Anonymous on-blockchain and off-blockchain bitcoin transactions," in *Financial Cryptography and Data Security: FC 2016 International Workshops, BITCOIN'16*. Springer Berlin Heidelberg, 2016, pp. 43–60.

[169] 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.

[170] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology: Proceedings of Crypto 82*. Springer US, 1983, pp. 199–203.

[171] 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.

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

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

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

[175] 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.

[176] G. Wood, "ETHEREUM: A secure decentralised generalised transaction ledger," *Available: http://gavwood.com/Paper.pdf*, Tech. Rep., 2014.

[177] J. R. Willett, "The second bitcoin," *Available: https://sites.google.com/site/2ndbtcwpaper/ 2ndBitcoinWhitepaper.pdf*.

[178] U. Mukhopadhyay, A. Skjellum, O. Hambolu, J. Oakley, L. Yu, and R. Brooks, "A brief survey of cryptocurrency systems," in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, Dec 2016, pp. 745–752.

[179] A. Kiayias, E. Koutsoupias, M. Kyropoulou, and Y. Tselekounis, "Blockchain mining games," in *Proceedings of the 2016 ACM Conference on Economics and Computation*, ser. EC '16. ACM, 2016, pp. 365–382.

[180] 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.

[181] B. Fisch, R. Pass, and A. Shelat, "Socially optimal mining pools," in *Web and Internet Economics: 13th International Conference, WINE 2017, Bangalore, India, December 17–20, 2017, Proceedings*, N. R. Devanur and P. Lu, Eds. Cham: Springer International Publishing, 2017, pp. 205–218.

[182] A. Kiayias, N. Lamprou, and A.-P. Stouka, "Proofs of proofs of work with sublinear complexity," in *Financial Cryptography and Data Security: FC 2016 International Workshops, BITCOIN, VOTING, and WAHC*. Springer Berlin Heidelberg, 2016, pp. 61–78.

[183] 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.

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

[185] J. Herrera-Joancomartí and C. Pérez-Solà, "Privacy in bitcoin transactions: New challenges from blockchain scalability solutions," in *Modeling Decisions for Artificial Intelligence: 13th International Conference, MDAI 2016*. Springer International Publishing, 2016, pp. 26–44.

[186] Y. Lewenberg, Y. Sompolinsky, and A. Zohar, "Inclusive block chain protocols," in *Financial Cryptography and Data Security: 19th International Conference, FC 2015*. Springer Berlin Heidelberg, 2015, pp. 528–547.

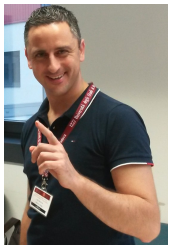
[187] G. Ateniese, B. Magri, D. Venturi, and E. Andrade, "Redactable blockchain – or – rewriting history in bitcoin and friends," 2016, http://eprint.iacr.org/2016/757.

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

[189] M. Milutinovic, W. He, H. Wu, and M. Kanwal, "Proof of luck: An efficient blockchain consensus protocol," in *Proceedings of the 1st Workshop on System Software for Trusted Execution*, ser. SysTEX '16. ACM, 2016.

[190] 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.

[191] G. Zyskind, O. Nathan, and A. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *2015 IEEE Security and Privacy Workshops*, May 2015, pp. 180–184.



Mauro Conti is an Associate Professor at the University of Padua, Italy.He obtained his Ph.D. from Sapienza University of Rome, Italy, in 2009.After his Ph.D., he was a Post-Doc Researcher at Vrije Universiteit Amsterdam, The Netherlands. In 2011 he joined as Assistant Professor the University of Padua, where he became Associate Professor in 2015. In 2017, he obtained the national habilitation as Full Professor for Computer Science and Computer Engineering. He has been Visiting Researcher at GMU (2008, 2016), UCLA (2010), UCI (2012, 2013, 2014), TUDarmstadt (2013), UF (2015), and FIU (2015, 2016). He has been awarded with a Marie Curie Fellowship (2012) by the European Commission, and with a Fellowship by the German DAAD (2013). His main research interest is in the area of security and privacy. In this area, he published more than 170 papers in topmost international peer-reviewed journals and conference. He is Associate Editor for several journals, including IEEE Communications Surveys Tutorials and IEEE Transactions on Information Forensics and Security. He was Program Chair for TRUST 2015, ICISS 2016,WiSec 2017, and General Chair for SecureComm 2012 and ACM SACMAT 2013.He is Senior Member of the IEEE.

Sandeep Kumar E completed his Bachelor of Engineering (B.E) in 2009 from Department of Telecommunication Engineering from JNN College of Engineering, Shimoga, Karnataka, India. In the same year he started his career as a faculty in the same department and same college. In 2014, he completed his Master of Technology (M.Tech) in Digital Communication Engineering from Department of Telecommunication Engineering, M.S Ramaiah Institute of Technology, Bangalore, Karnataka, India with first rank honors and gold medal. Presently he is working as Asst. Professor, in the Dept. Telecommunication Engineering, JNNCE, Shimoga, Karnataka, India. His current research areas include Blockchain Analysis, Security in Wireless networks, Software-defined networking, and Underwater acoustic networks.

Chhagan Lal is Postdoc fellow in Department of Mathematics, University of Padua, Italy. He obtained his Bachelors in Computer Science and Engineering from MBM Engineering College, Jodhpur, India in 2006. He obtained his Masters degree in Information Technology with specialization in Wireless communication from Indian Institute of Information Technology, Allahabad in 2009, and Ph.D. in Computer Science and Engineering from Malaviya National Institute of Technology, Jaipur, India in 2014. He has been awarded Canadian Commonwealth scholarship in 2012 under Canadian Commonwealth Scholarship Program to work in University of Saskatchewan in Saskatoon, Saskatchewan, Canada. His current research areas include Blockchain Analysis, Security in Wireless networks, Software-defined networking, Underwater acoustic networks, and context-based security solutions for Internet of Things (IoT) networks.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2842460, IEEE Communications Surveys & Tutorials

Sushmita Ruj (SM15) received the B.E. degree in computer science from Bengal Engineering and Science University, Shibpur, India, and the masters and Ph.D. degrees in computer science from Indian Statistical Institute, India. She was an Erasmus Mundus Post-Doctoral Fellow with Lund University, Sweden, and a Post-Doctoral Fellow with the University of Ottawa, Canada. She was an Assistant Professor with IIT Indore, Indore. She was a visiting researcher with INRIA, France, University of Wollongong, Australia, Kyushu University, Japan, the KDDI labs, Japan, and the Microsoft Research Lab, India. She is currently an Assistant Professor with Indian Statistical Institute. Her research interests are in applied cryptography, security, combinatorics, complex network analysis, blockchain, cryptocurrencies, mobile ad hoc networks, vehicular networks, cloud security, and security in smart grids. She has served as a program Co-Chair of IEEE ICCC (PS Track), IEEE ICDCS, and IEEE ICC, and served on many TPCs. She won the best paper awards at ISPA07 and IEEE PIMRC11. She is a Senior Member of the ACM.