Appendix A: Detailed Comments and Classification scheme

1 Introduction

1.1 What is a bitcoin?

It is nothing but a chain of transactions. It is a certificate which states what all transactions have taken place using this same bitcoin previously. The bearer of one such bitcoin can exchange this in terms of actual real world currency. Why is a bitcoin so popular nowadays? It works completely on a peer2peer network which makes regulation nearly impossible. It is unlike any other real world currency because it is immune to regulation. No real world government or institution can claim ownership of the entire network or concept. It is entirely market regulated and depends on the basic economic principle of demand and supply. It is a cryptocurrency and using an unprecedented amount of parallel computing technique, the entire network is able to ensure fraud proof operation. In other words it is nearly impossible to fool the network and peddle ones own fake certificates or fake bitcoins posing as the real ones in the network.

1.2 How does this work?

The cornerstone of the entire bitcoin network is a transaction. A transaction is nothing but transfer of a bitcoin from one owner to another. There could be a few potential issues with this. One is that somewhere down the line when the bitcoin has changed many owners, a malicious owner might try to double spend the bitcoin. In other words, he might try to sell the same bitcoin to two different owners at the same time. Obviously such a transaction should be illegal in the system as it has the potential to throw the entire network out of gear. To solve it, the bitcoin introduces the concept of block chain which in simple terms is a record of all previous transactions. Therefore when a transaction is about to happen, one can easily check

previous records to make sure there has been no double spending. This is done using a distributed timestamp server which is based on a peer to peer network, which would provide the necessary proof of previous transactions when required. An interesting aspect is provided in [15] which gives a good overview of Bitcoin user behavior and general quantitative characteristics of the Bitcoin network. On a related note, [19] propose an extension to the Bitcoin network that works on mutual trust but promises faster transaction time by moving it from a Pioneer model to a mutual trust based model.

1.3 Transactions

Each owner has his own public key and a private key. In a nutshell what happens when a bitcoin is passed on to another owner is that a new record is created in the block chain which has the hash of the previous transaction and the new owners public key as inputs. The result is appended to the end of the bitcoin thereby completing the transactions. But problem of double spending still persists. What if the owner, were to make a digital copy and use it twice. There must be a way to ensure that the previous owners did not sign any earlier transactions. Since the earliest transaction is the one that counts,later attempts to double spend can be neglected. If we are aware of all transactions we can be reasonably clear about the absence of any double spending operations. The mint is aware of all transactions and decides which arrived first. In order to completely do away with the concept of a thrid party validator all transactions must be publicly announced. This implies all participants must agree on a single history. To address this problem the requirement of a central mint is necessary. This mint will have the task of verifying each transaction and validating its correctness thereby preventing fraud. Mint is being implemented in a peer-2-peer model. But central mint is giving too much

Solution will be to establish a timestamp server. Each transaction will be timestamped. It is a unique way of making sure the amount of knowledge that was present at that exact point of time. This will help check whether any of the previous owners signed any of the previous documents. Each new transaction will take into account the previous timestamp and will include it in the hash to create the new timestamp for the new transaction.

Paper on [2] is taken as the backbone for implementing such a network. The technique of proof of work has been explained in detail in that work and is the basis for a similar technique being implemented in the bitcoin network too. Each block has a field called nonce. It is important to note that each transaction spawns a new block. The idea of the proof of work is that each block must hash and generate only

a certain number of zeroes. By manipulating the nonce field and by trial and error in an increasing fashion, a nonce value will be found with will give upon hashing a certain number of bits to the block. This will form the block. When after each successive transaction, the chain keeps getting augmented with new blocks with different nonce values depending on the nonce value of not just the current block but all previous blocks because as can be noted, each transaction also depends on previous transactions. The advantage of this process is that if an attacker were to manipulate the network and try to insert a wrong value, he would have to change the nonce of the entire chain henceforth which would prove to be computationally impractical.

The paper then talks about how unlikely and computationally hard it is for an attacker to replicate the same chain and keep pace with the network with intent to destroy it successfully. Since it is not in the interest of a parallel technique we can skip this part.

Paper further talks about incentivizing those nodes that lend the CPU time and electricity for facilatating the transactions and being part of the peer to peer network by generating new bitcoins for such nodes. The logic behind this is to prempt any attacker from using his computing powers to attack the system by offering him an even more profitable opportunity by help being part of the network and miniting new bit coins.

1.4 Block and block chain

The block is nothing but a transaction. Block chain is nothing but a public ledger which keeps track of all transaction in the bit coin network. *The blockchain is a public ledger of all transactions in the Bitcoin network. Blockchain.info allows you to navigate the bitcoin blockchain.*[3]. Thus when a transaction is performed the node handling the transaction must propagate this information and make sure that it is committed to other copies of the ledger. How this is done is explained in the next work.

2 Information Propagation in bitcoin network

[3] talks about various concepts like transactions, blocks and block chain that are being used in the bitcoin network. The work describes the implementation details of the p2p network. Since the purpose is to keep updating and synchronizing the ledger replicas, the relevant entities are the transaction and the block chain. Thus each node advertises with the inv message stating that it has these many blocks corresponding

to some transactions. Any node not having the said information responds back and transfer takes place. Paper talks about various concepts like transactions, blocks and block chain that are being used in the bitcoin network. The work describes the implementation details of the p2p network. Since the purpose is to keep updating and synchronizing the ledger replicas, the relevant entities are the transaction and the block chain. Thus each node advertises with the inv essage stating that it has these many blocks corresponding to some transactions. Any node not having the said information responds back and transfer takes place. This is done to save bandwidth as the information to be exchanged is of considerable size. The paper talks about blockchain forks and how they are created. Due to the concept of proof-of-work the valid blocks are to be found independently at random. The proof-of-work causes valid blocks to be found independently at random.

2.1 Where is parallel computing involved?

Starting from the beginning this is the sequence of events which goes on. An entity A wants to send BTCs to another entity B, and this is referred to as a transaction. This transaction typically is validated by using the techniques described in the previous sections. This transaction is *heard* by one of the nodes which picks it up creates a *block* and tries to publish it publicly i.e. to all other nodes in the network. But there are other such transactions being handled by other nodes simultaneously.The problem now is that how will a node decide which transaction happened first- the one which is represented by the block it received or the one that happened under its own watch. It is important to note that to keep the copy of the ledger fresh, only the latest transactions ordered in the correct chronological order must be put in. To do this, each node tentatively commits the transactions which it has knowledge of. If an earlier transaction is received as a broadcast from some other node then it is supposed to roll back the commit, put the latest information in and then push the remaining information in. This step though it looks simple actually is a bit more complicated than that. This problem refers to solve the proof of work problem referred to above. To determine the order the nodes attempt to find a solution to a proof-of-work. The proof-of-work consists in finding a byte string, called nonce, as illustrated in the previous section, that when combined with the block header has to yield a hash with a given number of zeroes. This problem is actually a computationally hard problem since cryptographic hash functions are one-way functions. To find the actual nonce string we have to analyze all possibilities which would give us the correct value. Once we obtain the correct value, it is easy to verify its correctness. The node finding the nonce value first will then send the block after embedding it with the nonce value to

all other nodes. Since it is easy to verify the authenticity, they will then accept or reject the solution and accordingly make changes to their own copies of the ledger. This aspect of the network is what is the most powerful feature of the network.

2.2 Blockchain

Having covered the concept of blocks and transactions, we now come to the concept of block chains. Blockchains are nothing but a directed tree with individual blocks for nodes with the latest node being referred to as the block chain head. The height of the tree is referred to as h. There can be a situation in which different blockchain heads can exist at different heights. In such a scenario, if a node receives a block with a height greater than the block chain height of its own ledger copy, there can be two cases, one in which the existing block chain head is an ancestor of the received block or when its not. If it is the ancestor, it is obvious that the node has missed out on some transactions in between and will get the required information from other nodes and attempt to keep its copy of the ledger fresh. If however it is not an ancestor, it is clear that, both share a common ancestor, and the node will then find refreh the list starting from the nearest common ancestor to keep its copy fresh.

3 Keys:Maintaining Privacy in a distributed setting

A necessary aspect in the scenario for Bitcoins is the need to maintain privacy but at the same time ensuring that safety in the transaction. This is particularly important because we need to maintain the principles of authenticity,integrity and non repudiation while implementing the Bitcoin network. Authenticity is the ability to identify the sender of the message. Integrity implies that the message being sent must not be compromised by a malicious third party. Any person in the middle must not be able to significantly alter or replace a part of the message being sent. Non repudiation means that the sender must not be able to refute the sending of the message sometime in the future after it has been received by the receiver [18] . It is in this regard that cryptographic concepts of hash function is used in this network. hash functions provide the much needed support in the Bitcoin network with the use of two sets of keys i.e. the public and private keys. These keys accomplish the dual purpose of concealing the identity of the individual involved in the transaction while ensuring the safety of the transactions. Since the ledger is public and scattered across a p2p network, this is accomplished among the many nodes in

a distributed way. It is vital to the network to avoid fraud or theft of coins. This section focusses on methods which ensure authenticity,integrity and non-repudiation in a Bitcoin transaction between two interested parties without the involvement of any third entity.

3.1 Public key cryptography

Public keys also serves as a method for verifying the authenticity and the integrity because with the public key one can decrypt the signature and and compare the result with the hash of the message. This also implements non repudiation as the sender is not able to falsely deny the sending of the message. The paper [5] provides more in-depth detail about public key encryption systems and the motivation behind it and is considered a seminal paper in the said domain.

3.2 Hashing:A basic overview

If two parties want to send or receive messages, they can use encryption to hide the messages. The receiver can then perform what is referred to as decryption to recover the original message. It is often the case that the algorithms for encryption and decryption are well known and the receiver can recover the original message using what is referred to as a key. If the key is the same as the one used for encryption, it is referred to as a *symmetric encryption*. The advantage of symmetric algorithms lies in the aspect of confidentiality, but the process for maintaining a common key between sender and receiver is often cumbersome. This is the motivation for use of hash functions. A hash function akes a variable-length input string referred to as a pre-image and converts it to a fixed-length output string called a hash value.[18].

3.3 Protocols for Public key encryption

The paper [11] provides some established protocols which deal with exchange of keys in a distributed environment. In [11] is highlighted several techniques relating to key distribution. The first one among them is the centralized key distribution technique which employs a central entity or a distribution center which serves as a repository for all the agents to deposit their respective keys. If any two agents wish to communicate with each other they contact the distribution center and obtain the keys. In case of the Bitcoin network however, such an approach is not feasible because, the entire network design being a peer-2-peer one. he work presented in [14] gives a detailed description of methods for obtaining digital signatures and public-key crypto systems.

It proposes a scheme which involves an exponential rate hash function. [4] presents another work which talks about vulnerabilities in public key crypto systems and goes on to describe a method to foil attacks arising out of such vulnerabilities. The author argues that an attacker can get the victim to sign new messages derived by intercepting messages to the victim and then forging the signature of the victim. [13] and [6] provide description of various types of probabilistic cryptographic techniques and an approach to enhance security in public key encryption systems by preventing cipher text attacks.

4 Timestamping:An extension to proof-of-work

In Section 1.3 we provided an overview of the timestamping process which goes on in the network. In this section we deal with a deeper analysis of the concept of timestamping and its relevance to the Bitcoin network. This section describes approaches regarding timestamping in situations wherein the trust factor is distributed i.e. in cases where there is no centralized authority to guarantee trustworthiness of a transaction.

4.1 Why is it a necessity?

Timestamping basically is a measure to prevent double spending. By timestamping a certain transaction we guarantee the temporal aspect of the transaction. In the Bitcoin network, it is also essential to keep track of previous transactions too to instill legitimacy in the transaction. This section talks about some approaches which can be used for timestamping as referred to by [17].

4.2 Timestamping: A general outline

[10] [8] In a system where there is a centralized system of authority, [10] propose a technique wherein a timestamping scheme following the binary tree structure elaborated in [8] is used and an improved scheme with minimum trust requirements. One method to link the present timestamp using an appropriate hash function to the previously found timestamp. As this chain grows, it becomes increasingly difficult to for any attacker to forge timestamps by manipulating the bit-strings[7].

4.3 Timestamping:Relation to proof-of-work

The original paper makes a reference to [2] as the prototype for implementing a distributed crypto currency network like Bitcoin. Hashcash [2] was originally intended to be used as a throttling system for unwarranted use of internet resources like email spam. It basically consists of a client intending to take part in a protocol to fulfill certain computation tasks and generate a coin in order to be eligible for consideration by the server. It proposes the use of a cost function which is intended to be easily verifiable but expensive to compute.

5 Pooling and the Bitcoin Reward system

The paper [16] also lists out different pooling strategies or the ways in which the reward distribution takes place. There are two techniques for going about mining for bit coins. One of them is solo mining and one of them is pooled mining. In the following subsections we will be discussing the importance of pooled and solo mining.

5.0.1 Solo mining

If mining for t time results in a $\frac{ht}{2^{32}D}$ blocks on average then we can say that $\lambda = \frac{ht}{2^{32}D}$ which is also the variance of the number of blocks found. This means that for the payout the following is the variance[16]:

5.0.2 Pooled mining

Pooled mining is a technique when a group of miners join together and collectively try to find the next block header. If we consider H as the hash rate of all the miners together, then total average reward is $\frac{Ht}{2^{32}D}$. An individual miners share is q the he will have qH of the share of the total hashes. This means his reward is $q\frac{Ht}{2^{32}D}$ which is nothing but $\frac{ht}{2^{32}D}$. However his payout variance is now very small $q\frac{htB^2}{2^{32}D}$, which means that The potential benefit to the miner is greater if the miner is small and the pool is large.

In [1] some there techniques of rewards have been discussed which improve upon the reward mechanism and attempt to lessen the payment overhead plaguing current techniques.

6 Disadvantages and alternatives

As we have seen in the previous sections, the crux of the whole Bitcoin network lies in the finding of a block by nodes by solving a computationally hard problems. This requires intense use of computing resources which consume significant amount of resources in terms of electricity, CPU time and ultimately money. Another aspect of Bitcoin network is the apparent increase in difficulty level as the network becomes popular leading to more nodes joining the network introducing greater competition to existing nodes.

As a result many alternatives like Memcoin and Litecoin [9] have been proposed. These approaches use a sequential memory hard scheme which require more memory than normal schemes mentioned in previous sections. Another alternative known as Zerocoin [12]has been proposed which is an extension to the Bitcoin network and seeks to fully anonymize Bitcoin transaction without significantly altering the network internals.

7 Conclusion and Future Scope

References

[1] Moshe Babaioff, Shahar Dobzinski, Sigal Oren, and Aviv Zohar. On bitcoin and red balloons. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, pages 56–73, New York, NY, USA, 2012. ACM.

[2] Adam Back. Hashcash - a denial of service counter-measure. 2002.

[3] C. Decker and R. Wattenhofer. Information propagation in the bitcoin network. In *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, pages 1–10, Sept 2013.

[4] Dorothy E. Denning. Digital signatures with rsa and other public-key cryptosystems. *Commun. ACM*, 27(4):388–392, April 1984.

[5] W. Diffie and M.E. Hellman. Privacy and authentication: An introduction to cryptography. *Proceedings of the IEEE*, 67(3):397–427, March 1979.

[6] Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. In *Public Key Cryptography*, volume 1560 of *Lecture Notes in Computer Science*, pages 53–68. Springer Berlin Heidelberg, 1999.

[7] Stuart Haber and W. Scott Stornetta. How to time-stamp a digital document. *Journal of Cryptology*, 3:99–111, 1991.

[8] Stuart Haber and W. Scott Stornetta. Secure names for bit-strings. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, CCS '97, pages 28–35, New York, NY, USA, 1997. ACM.

[9] Adam Mackenzie. Memcoin2: A hybrid proof of work/proof of stake cryptocurrency. 2010.

[10] H. Massias, X. Serret Avila, and J.-J. Quisquater. Design of a secure timestamping service with minimal trust requirement. In *the 20th Symposium on Information Theory in the Benelux*, 1999.

[11] Ralph C. Merkle. Protocols for public key cryptosystems. *2012 IEEE Symposium on Security and Privacy*, 0:122, 1980.

[12] Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, SP '13, pages 397–411, Washington, DC, USA, 2013. IEEE Computer Society.

[13] Tatsuaki Okamoto, Shigenori Uchiyama, and Eiichiro Fujisaki. Epoc: Efficient probabilistic public-key encryption. In *IEEE P1363a*, 1998.

[14] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978.

[15] Dorit Ron and Adi Shamir. Quantitative analysis of the full bitcoin transaction graph. In Ahmad-Reza Sadeghi, editor, *Financial Cryptography and Data Security*, volume 7859 of *Lecture Notes in Computer Science*, pages 6–24. Springer Berlin Heidelberg, 2013.

[16] Meni Rosenfeld. Analysis of bitcoin pooled mining reward systems. *CoRR*, abs/1112.4980, 2011.

[17] Satoshi. Bitcoin: A peer-to-peer electronic cash system. 2008.

[18] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, Inc., New York, NY, USA, 1993.

[19] P. Singh, B.R. Chandavarkar, S. Arora, and N. Agrawal. Performance comparison of executing fast transactions in bitcoin network using verifiable code execution. In *Advanced Computing, Networking and Security (ADCONS), 2013 2nd International Conference on*, pages 193–198, Dec 2013.

Appendix B: Annotated Bibliography

Paritosh P. Ramanan

The system of a crypto currency has been suggested [16] which attempts to restrict the transaction only between two responsible parties and without depending on trust in facilitating the transaction. Cryptocurrency like Bitcoin works on the basis of cryptographic proof instead of trust. It removes the concept of the middle third party by directly engaging the buyer and seller. The premise of the crytocurrency is to develop a technique wherein a transaction once made is computationally too impractical to reverse. Cryptocurrency is analogous to legal tender or hard currency in the digital world. The analogy stems from the fact that like legal tender which once used in a transaction cannot be reversed theoretically. The purpose of Bitcoin is to replicate this mechanism prevalent in case of legal tender in the electronic domain which would allow for a seamless transaction of money between two individual removing the issues of trust and other logistical disadvantages.

It works completely on a peer2peer network which makes regulation nearly impossible. It is unlike any other real world currency because it is immune to regulation. No real world government or institution can claim ownership of the entire network or concept. It is entirely market regulated and depends on the basic economic principle of demand and supply. It is a cryptocurrency and using an unprecedented amount of parallel computing technique, the entire network is able to ensure fraud proof operation. In other words it is nearly impossible to fool the network and peddle ones own fake certificates or fake bitcoins posing as the real ones in the network. An interesting aspect is provided in [14] which gives a good overview of Bitcoin user behavior and general quantitative characteristics of the Bitcoin network.

Since the earliest transaction is the one that needs to be examined, later attempts to double spend can be neglected. If we are aware of all transactions we can be reasonably clear about the absence of any double spending operations. The mint is aware of all transactions and decides which arrived first. In order to completely do away with the concept of a thrid party validator all transactions must be publicly announced. This implies all participants must agree on a single history. On a related note, [18] propose an extension to the Bitcoin network that works on mutual trust but promises faster transaction time by moving it from a Pioneer model to a mutual trust based model.

Paper on [1] is taken as the backbone for implementing such a network. The technique of proof of work has been explained in detail in that work and is the basis for a similar technique being implemented in the bitcoin network too. Each block has a field called

nonce. It is important to note that each transaction spawns a new block. The idea of the proof of work is that each block must hash and generate only a certain number of zeroes. By manipulating the nonce field and by trial and error in an increasing fashion, a nonce value will be found with will give upon hashing a certain number of bits to the block. This will form the block. When after each successive transaction, the chain keeps getting augmented with new blocks with different nonce values depending on the nonce value of not just the current block but all previous blocks because as can be noted, each transaction also depends on previous transactions. The advantage of this process is that if an attacker were to manipulate the network and try to insert a wrong value, he would have to change the nonce of the entire chain henceforth which would prove to be computationally impractical.

[2] talks about various concepts like transactions, blocks and block chain that are being used in the bitcoin network. The work describes the implementation details of the p2p network. Since the purpose is to keep updating and synchronizing the ledger replicas, the relevant entities are the transaction and the block chain. Thus each node advertises with the inv message stating that it has these many blocks corresponding to some transactions. Any node not having the said information responds back and transfer takes place. This is done to save bandwidth as the information to be exchanged is of considerable size.

Authenticity is the ability to identify the sender of the message. Integrity implies that the message being sent must not be compromised by a malicious third party. Any person in the middle must not be able to significantly alter or replace a part of the message being sent. Non repudiation means that the sender must not be able to refute the sending of the message sometime in the future after it has been received by the receiver [17] . It is in this regard that cryptographic concepts of hash function is used in this network. hash functions provide the much needed support in the Bitcoin network with the use of two sets of keys i.e. the public and private keys. These keys accomplish the dual purpose of concealing the identity of the individual involved in the transaction while ensuring the safety of the transactions. The paper [4] provides more in-depth detail about public key encryption systems and the motivation behind it and is considered a seminal paper in the said domain. The paper [10] provides some established protocols which deal with exchange of keys in a distributed environment. A hash function akes a variable-length input string referred to as a pre-image and converts it to a fixed-length output string called a hash value.[17]. The work presented in [13] gives a detailed description of methods for obtaining digital signatures and public-key crypto systems. It proposes a scheme which involves an exponential rate hash function. [3] presents another work which talks about vulnerabilities in public key crypto systems and goes on to describe a method to foil attacks arising out of such vulnerabilities. The author argues that an attacker can get the victim to sign new messages derived by intercepting messages to the victim and then forging the signature of the victim. [12] and [5] provide description of various types of probabilistic cryptographic techniques and an approach to enhance security in public key encryption systems by preventing cipher text attacks.

There are certain conditions that need to be implemented to achieve a stable system

of verifying against forgery of timestamping. Thus, there have been recent attempts to develop timestamping schemes which will be computationally very hard to fake. One such method is to link the present timestamp using an appropriate hash function to the previously found timestamp. As this chain grows, it becomes increasingly difficult to for any attacker to forge timestamps by manipulating the bit-strings[6].

The timestamping scheme being used in the Bitcoin network is inspired from the works [7] and [9].

The timestamp of the document now contains all the values necessary to rebuild the entire tree. For instance, for y_3 the timestamp is $\{(y_3, L), (H_12, L), (H_58, R), (RH_i - 1, L)\}$. It is tone noted that this notation basically comprises of the left or the right sibling denoted by letter R or L and their hash values respectively. The general idea is that for verification purposes the "Round value" is obtaining by systematically rebuilding the tree using the above mentioned information. The binary tree structure we are using in this method is referred to as the Merkle tree[10].

A user initially obtains the block header of the longest proof-of-work chain which can be easily obtained by querying the network. Then one can obtain the Merkle root by linking the transaction to the block its timestamped in. By linking the transaction to a particular place in the chain, it becomes obvious that a network node has accepted it and the blocks added after it can be used as proof that the network has accepted it. The authors in [16] argue that as long as a majority of nodes in the network are honest nodes, the system will be fraud free as an attacker would have to be able to generate an even longer chain which is computationally impossible to achieve unless the majority of nodes are influenced to perpetrate the attack.

The paper [15] also lists out different pooling strategies or the ways in which the reward distribution takes place. There are two techniques for going about mining for bit coins. One of them is solo mining and one of them is pooled mining. In the following subsections we will be discussing the importance of pooled and solo mining. In [**?**] some there techniques of rewards have been discussed which improve upon the reward mechanism and attempt to lessen the payment overhead plaguing current techniques.

As a result of all these factors many alternatives like Memcoin and Litecoin [8] have been proposed. These approaches use a sequential memory hard scheme which require more memory than normal schemes mentioned in previous sections. Another alternative known as Zerocoin [11]has been proposed which is an extension to the Bitcoin network and seeks to fully anonymize Bitcoin transaction without significantly altering the network internals.

References

[1] Adam Back. Hashcash - a denial of service counter-measure. 2002.

[2] C. Decker and R. Wattenhofer. Information propagation in the bitcoin network. In

Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on, pages 1–10, Sept 2013.

[3] Dorothy E. Denning. Digital signatures with rsa and other public-key cryptosystems. *Commun. ACM*, 27(4):388–392, April 1984.

[4] W. Diffie and M.E. Hellman. Privacy and authentication: An introduction to cryptography. *Proceedings of the IEEE*, 67(3):397–427, March 1979.

[5] Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. In *Public Key Cryptography*, volume 1560 of *Lecture Notes in Computer Science*, pages 53–68. Springer Berlin Heidelberg, 1999.

[6] Stuart Haber and W. Scott Stornetta. How to time-stamp a digital document. *Journal of Cryptology*, 3:99–111, 1991.

[7] Stuart Haber and W. Scott Stornetta. Secure names for bit-strings. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, CCS '97, pages 28–35, New York, NY, USA, 1997. ACM.

[8] Adam Mackenzie. Memcoin2: A hybrid proof of work/proof of stake cryptocurrency. 2010.

[9] H. Massias, X. Serret Avila, and J.-J. Quisquater. Design of a secure timestamping service with minimal trust requirement. In *the 20th Symposium on Information Theory in the Benelux*, 1999.

[10] Ralph C. Merkle. Protocols for public key cryptosystems. *2012 IEEE Symposium on Security and Privacy*, 0:122, 1980.

[11] Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, SP '13, pages 397–411, Washington, DC, USA, 2013. IEEE Computer Society.

[12] Tatsuaki Okamoto, Shigenori Uchiyama, and Eiichiro Fujisaki. Epoc: Efficient probabilistic public-key encryption. In *IEEE P1363a*, 1998.

[13] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978.

[14] Dorit Ron and Adi Shamir. Quantitative analysis of the full bitcoin transaction graph. In Ahmad-Reza Sadeghi, editor, *Financial Cryptography and Data Security*, volume 7859 of *Lecture Notes in Computer Science*, pages 6–24. Springer Berlin Heidelberg, 2013.

[15] Meni Rosenfeld. Analysis of bitcoin pooled mining reward systems. *CoRR*, abs/1112.4980, 2011.

[16] Satoshi. Bitcoin: A peer-to-peer electronic cash system. 2008.

[17] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, Inc., New York, NY, USA, 1993.

[18] P. Singh, B.R. Chandavarkar, S. Arora, and N. Agrawal. Performance comparison of executing fast transactions in bitcoin network using verifiable code execution. In *Advanced Computing, Networking and Security (ADCONS), 2013 2nd International Conference on*, pages 193–198, Dec 2013.