Rep on the Roll: A peer to peer reputation system based on a rolling blockchain

Richard Dennis
School of Computing
University of Portsmouth
Portsmouth, United Kingdom
Richard.dennis@port.ac.uk

Gareth Owenson
School of Computing
University of Portsmouth
Portsmouth, United Kingdom
Gareth.owenson@port.ac.uk

***Abstract*— This paper presents the first generalized reputation system that can be applied to multiple networks that is based on the blockchain. We first discuss current reputation systems, conducting a critical analysis of their current security vulnerabilities, before looking at how new blockchain-based technologies are used. We propose an innovative new reputation system that is based on blockchain technologies and which aims to solve many unanswered questions in the current generation of reputation systems, as well as blockchain systems in general. We then consider the limitations of such a system, before using simulations and analyses to demonstrate methods of overcoming these limitations, and in doing so, provide a possible solution to a fundamental issue in blockchain-based networks; scalability. We conclude by suggesting areas for future studies, and summarizing our findings.**

Blockchain, scalability, reputation systems, cryptographic protocols, distrubted networks, peer-to-peer, Bittorent

I. INTRODUCTION

This is an extension of the paper titled "Rep on the block: A next generation reputation system based on the blockchain."

Reputation measures how much the community trusts you, and is calculated on your previous transactions and interactions with the community. The greater your reputation, the more trustworthy you are seen to be on the network and, with a user's reputation on the line, users choose to behave more honestly on the network [1].

At present, eBay has the most widely used reputation system and processes over a billion transactions per day [2]. Each transaction could result in two reputation scores being left (one from the buyer, the other from the seller); it is therefore essential that reputation systems can handle a large number of transactions, and have adequate sources to handle this level of data.

E-commerce reputation systems often implement multi-dimensional reputations which allow the user to rate the seller on a range of factors such as postage cost and quality of communications. All major E-commerce websites use the traditional client-server model, where the reputation data is centrally stored, calculated and distributed on a centralized server and all clients can request to see this data from the central server.

In eBay's system, the positive feedback percentage is calculated based on the total number of positive and negative feedback ratings for transactions in the last 12 months, excluding repeat feedback from the same member for purchases made within the same calendar week [3].

The reputation score is calculated centrally by the E-commerce website, resulting in the company being able to change the reputation calculation algorithm and force its deployment to all users without their knowledge. A recent example of this is when eBay began preventing sellers from leaving negative feedback about buyers.

Although successful reputation systems have been implemented on multiple web services, they are all based on the centralised server model, which makes them unsuitable for deployment in peer-to-peer (P2P) networks whose main purpose is to decentralise control away from a single entity. Thus far, the effective communication and sharing of unmodified information relating to trust and reputation remains an unsolved issue [4].

There are several reputation systems implemented in peer-to-peer networks which aim to provide users of the network with an incentive to behave honestly and to deter "freeloaders". Freeloaders are peers who download content from the network, but who do not distribute any content. It has been estimated that on the Gnutella network - the most popular P2P network - approximately 70% of all peers can be considered freeloaders [5].

There are various implementations of reputation systems on peer-to-peer networks; some require the implementation of a trusted central server, much like the E-commerce model, which records and calculates all users' ratings, whilst other systems try to distribute the reputation system with a distributed database that all peers on the network have an updated copy of. The final implementation of reputation systems on a P2P network only records reputations of peers it has interacted with.

Unlike E-commerce reputation systems where participation is mandatory, enrolment in a P2P reputation system is optional and many nodes are concerned about the loss of privacy or the additional resources that are required.

P2P reputation systems are single-dimensional systems, with each peer only leaving one bit of data about the

transaction that has taken place; this enhances efficiency and also reduces load on the network.

The calculation of reputation differs from implementation to implementation, however the general calculation method for each peer is that their reputation is the sum of all reputation feedback received.

All reputation systems, no matter how they are deployed or what type of network they are deployed over, face the same fundamental issues. The ability to link an identity to a single user and to prevent that user from obtaining more than one identity is key to preventing a user exploiting the system by creating multiple identities and transacting between them.

Creators of reputation systems must always consider how reputation should be quantified – a question for which there is not currently a definitive answer. Furthermore, how can we ensure the reputation left by a user is accurate and is based on a real transaction?

Finally, all blockchain-based networks have not properly addressed the issue of scalability; which causes the original decentralized nature of the blockchain to become more centralized, with only the highest resourced users being able participate in the network. It remains to be seen whether it is possible to reduce the size of the blockchain in a secure and distributed manner, and if so, how this can be successfully implemented in all blockchain-based networks.

The structure of this paper is as follows: section two describes related work in this area, focusing on reputation systems implemented in peer-to-peer systems as well as scalability issues faced by blockchain-based networks. Section three discusses our proposed reputation system along with some of the technologies used in it, whilst section four summarises our approach with regards to simulation and comparison of our network to currently implemented reputation systems. In section five we consider the limitations of the proposed network, before focusing on solving the scalability limitation of this, and all, blockchain-based networks by reducing the blockchain size 92%. We then conclude with suggestions for future work and summarise the contribution of this paper.

II. Related Work

A. *Existing "decentralized" reputation systems*

Reputation systems are not just implemented to prevent freeloading on P2P systems. Other goals include: filtering out non-authentic files (pollution) on 2P networks, ensuring network resources are correctly selected and allocated, finding a method with which to identify high quality contributors to the network and a way to punish, or prevent, nodes from behaving dishonestly on the network.

Gupta et al. [5] presented one of the first reputation systems of peer-to-peer networks; this concept now forms the basis that several other reputation systems have been built upon. This novel system works on an "opt-in" basis, thus allowing users not to participate for privacy reasons. This approach is strongly criticized by Schiffner et al. [6] who acknowledge the need for privacy and anonymity in a reputation system, but believe that, for a system to work and to prevent attackers from undermining the system, all its users must participate in it.

Wang and Vassileva [7] also propose a reputation system which is based on the Bayesian model, and which aims to quantify the trust in each peer and the quality of files they share. Like the reputation system proposed by Gupta et al. [5], the nodes only gain a local view of the network, reputations are collected by each peer based on previous transactions and it relies on the nodes being honest when sharing this information. In reality, this assumption is unrealistic and would likely not hold true.

The reputation systems proposed by Gupta et al. [5], Schiffner et al. [6] and Wang and Vassileva [7] all use a binary rating score, which only allows positive values, with each successful transaction gaining a reputation score of 1, and each peers' reputation is the sum of these scores. There are several issues with this type of reputation. Perhaps the major issue is the assumption that all scores are genuine for a transaction that actually took place and that there are no malicious actors trying to profit from the system. Schiffner et al. [6] and Wang and Vassileva [7] do not attempt to address this matter, however Gupta et al. [5] propose a receipt-based system. In their system, users can request a receipt from the peer in question in order to show that their reputation score is from genuine transactions. It should be noted, however, that this method does not prevent two nodes from colluding together and sending genuine transactions between each other to increase their reputation scores.

The systems proposed by Wang and Vassileva [7] and Gupta et al. [5] fail to address both the issue of identity management – to ensure users can only obtain a single identity – and the possibility that peers may collude together in order to profit from the system to increase their own reputations. The design proposed by Kamvar et al. [8], however, enables the centralized server to conduct basic identity management – ensuring, for example, that there are not multiple identities based on a single IP address.

Several different kinds of reputation systems have been discussed in literature, and while some solve some problems, there is not yet a system that can solve all the issues faced in implementing a decentralized reputation system. The system proposed by Gupta et al. [5] is the most complete and effective, although it has some significant drawbacks that, once solved, will be able provide a truly decentralized reputation system over peer-to-peer networks.

B. *Attacks on decentralised reputation systems*

Reputation systems both on centralized networks and decentralized systems are ripe for attack, with significant financial benefits; e-commerce websites have demonstrated that users with a high reputation can expect to receive an 11.2% premium on all goods they sell [9].

The slandering attack, first described by Hoffman et al. [10], is perhaps the easiest attack to conduct. Attackers manipulate the reputation of other nodes by reporting ratings that do not reflect their genuine opinion in order to lower their reputation. Jøsang and Golbeck [11] describe a possible defence against such an attack by comparing ratings of users

to ratings left by more trusted users on the network, however this would allow highly trusted users to abuse their status in the community and conduct a slandering attack undetected. Evidence shows the slandering attack to have been conducted at state level by GCHQ in an attempt to discredit selected targets [12].

The Sybil attack, which was first described by Douceur [13], is an attack that all peer-to-peer networks are vulnerable to; it does not just affect reputation systems. The Sybil attack can be described as an attacker "legally" gaining more than a single identity. Hoffman et al. argue that the Sybil attack is the most important one to defend against, as it forms the foundation of nearly all attacks on reputation systems – the reason for this being attributed to the availability of cheap anonymous or pseudonymous identities [10]. Douceur [13] further elaborates on this, describing how the success of a Sybil attack depends on the cost of obtaining an identity, and clearly showing how the effectiveness of a Sybil attack is reduced when the cost of generating a new identity increases.

Both Danezis and Mittal [14] and Yu et al. [15] describe how the most effective countermeasure to the Sybil attack is to link identities on the network to a real world identity. This has been shown to almost entirely prevent a Sybil attack, although the cost to the network in terms of the resources required to verify each user is high, and the process needs to be done by a human as it cannot yet be automated. This makes the solution one that would not be scalable for millions of users.

The majority of distributed reputation systems do not allow users to accumulate negative reputation scores due to the re-entry attack. This attack exploits the cost of entry to a network; for users who are behaving maliciously, once their reputation impacts their attack, it is cheaper for them to stop using that account and recreate an account, than it is to regain positive reputation - this method is then repeated for the duration of the attack. Prêtre [16] rightly appraises this attack as efficient not only because of the low cost of entry to the network, but also because the network sees a user with zero reputation scores as higher than a user with negative scores, thus providing the user with an incentive to dispose of the account.

While the majority of reputation systems currently deployed are vulnerable to these - and more - attacks, Jøsang and Golbeck [11] question whether it is necessary for the reputation system to be perfectly secure. They argue that, in the majority of situations, there is little incentive to attack the network, and the value of a reputation system lies elsewhere.

Since P2P reflects society better than other types of computer architectures [17], it could also be argued that, combined with a reputation system, the majority of users would behave honestly – as they do in society.

C. Scalability issues with blockchain-based networks

Poon and Dryja [18] describe the Blockchain Scalability Problem as not being a single problem, but rather the combination of multiple issues that ultimately affect the possible scalability of the blockchain.

On average, VISA handles around 2,000 transactions per second (tps), with a recorded daily peak rate of 4,000 tps. It has a peak capacity of around 56,000 transactions per second [19]. By comparison, the maximum number of transactions per second that Bitcoin can currently theoretically achieve with the 1MB block size limit is 7 [20]. Poon and Dryja [18] describe how, whilst it is possible to achieve the tps VISA is capable of on Bitcoin, this would result in 8GB blocks, and a blockchain that would increase in size by over 400 terabytes a year.

Eyal et al. [21] doubt whether an increase in the block size can solve the scalability issue; their research paper demonstrates that, as an increased block size results in additional resources being required, an increased block size would mean fewer home computers would be able to participate in the network, and this would ultimately lead to centralization.

In his whitepaper, Nakamoto [22] states that the requirement to store all transaction history since the first transaction will ultimately result in the blockchain protocol failing to scale. Nakamoto foresaw a scalability issue during the design of Bitcoin and proposed a "pruning" mechanism. This mechanism allows a user to remove all spent transactions from their copy of the blockchain. A spent transaction is a transaction that can no longer be used as an input for a new transaction. However, this method has been criticized for still requiring a user to download the entire blockchain before they can start pruning and still requiring the majority of the nodes on the network to process a complete and unpruned blockchain [21].

III. OUR APPROACH

We propose a general blockchain-based reputation system that aims to solve several major challenges that the previous generations of reputation systems have failed to resolve, as well as preventing attacks that are possible on current generation reputation systems. We will focus on the application of this system on a peer-to-peer network, although it is also just as easily deployed on a classic E-commerce website.

Blockchain technology is a novel peer-to-peer approach to linking a sequence of transactions or events together in a way that makes them immutable. This was originally described by Nakamoto and implemented for the virtual currency Bitcoin [22]. In Bitcoin, users exchange money using transactions much like in real life. When a user creates a transaction he broadcasts this to all peers in the network. A special group of peers, called miners, collect broadcast transactions and attempt to incorporate them into a block that satisfies a cryptographic hash function. The process of producing a block is computationally intensive and probabilistic. Given a proposed block, each miner has a fixed and independent probability of successfully producing a block which satisfies the hash function for each unit of computation time. Whilst producing a block is hard, verification of a correct block is not.

Blocks are also linked together by chaining the hash of the previous block with each subsequent one. Thus, an attacker must control a significant proportion of the computation

power (typically 51%) to produce one false block and faking transactions back into the past is exponentially hard.

In Bitcoin, the collection of blocks (and their transactions) is called the ledger, and this is publicly inspectable by any peer. Thus a peer can see and verify any transaction from any point in time.

The blockchain was first described by Nakamoto in his paper describing the Bitcoin protocol [22]. The blockchain is a public ledger of all transactions that have ever been completed since the first "genesis" block. Each transaction from the Bitcoin protocol is broadcast to all nodes in the network which are maintaining the blockchain, known as miners.

These miners check the transactions were valid (e.g., sender has enough coins to send) and then package all the valid transactions into a block. All nodes have a complete copy of the blockchain and keep this up to date. The block must contain a cryptographic hash of the previous block, this is the method used to cryptographically link every block in the blockchain to its previous block, all the way back to the first, genesis block. Once the block has been assembled, all miners on the network undertake a challenge of finding a nonce, so that the hash of the current block contains a set amount of zeros at the start. This process is commonly referred to as mining. Mining is a competition between all miners on the network, and the first miner to find the nonce and publish this confirmed block to the network receives a set amount of Bitcoins.

The use of previous hashes in each block prevents any attack where the contents of a block is changed, as if this were to happen that block and all subsequent blocks hashes would not match up. The only method a user would be able to use to change data in a previous block is to control 51% of all computational power on the network. Known as the 51% attack, this attack requires a majority of the computational power to be used to "re-mine" each block from the block that was altered. This would require a substantial amount of computing power, as the Bitcoin network currently has 510,000,000 GH/S [23] of computational power solely dedicated to mining, which is 256 times more powerful than the combination of the top 500 supercomputers in the world [24].

It is this property that makes the blockchain into a very secure ledger, which will remain secure to all adversaries who control less than 51% of the computational power of the network, as the cost of resources required to control 51% would outweigh the potential rewards.

IV. DESCRIPTION OF OUR APPROACH

We propose a new reputation system based on blockchain technology. To reduce load on the current Bitcoin blockchain and to reduce inflation of the blockchain, we will create an entirely new blockchain, the sole purpose of which is to store reputation from completed transactions.

The proposed network has two goals – to withstand previously documented attacks on reputation systems and to provide a generalised reputation system that can be implemented into any network.

In a peer-to-peer network environment, we propose to solve the issue of quantifying reputation by removing the human opinion from the transaction. Our system will only store single dimensional reputation, with each user leaving either a 1 for a positive transaction, or a 0 for a non-satisfactory transaction. A positive transaction is classified as a transaction in which the user received the file they requested.

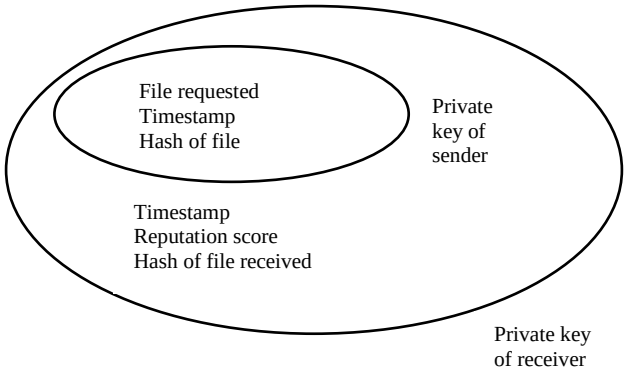

Figure 1*: Receipt of transaction sent to the miners*

We classify a transaction as the sending a piece of data, such as a file, signed by the sender's private key to a user who requested it.

Upon receiving the correct file, the user sends a transaction consisting of the reputation score, a timestamp, and a hash of the received file. This data is then encrypted with the receiver's private key and is sent to the miners. This ensures the reputation left by a user is based on a real transaction, a major issue in current generation reputation systems. The unfair ratings attack is now no longer possible since there is now cryptographic proof the user sent a requested file, and the user received it.

Fig 1 is a diagram of the format of a transaction which would be sent to the miners

The miners check the validity of the transaction by contacting each user involved in the transaction, and requests a signed proof, containing the file hash and a random nonce sent by the miner to be included. This is to prove each user sent/received the file, however this does have the drawback of requiring the users to still be online for the miners to verify the transaction. The miners then assemble these verified transactions into a block of other transactions before confirming them in a method identical to current Bitcoin implementation. Fig 2 shows some pseudo code detailing how a miner would verify a transaction.

A method to ensure users cannot generate multiple identities cheaply is to link the indemnity creation to the IP address of a user. IPV4 addresses are becoming more expensive to purchase, as there is a lack of them available. While this method does not prevent an attacker from creating multiple identities, it makes the cost of doing so much more expensive, thus deterring all but the most well-funded attacker.

Identity-based encryption systems with the ability to generate a public key based on an email address were also

evaluated and tested; this was a desirable feature, however the requirement of a centralized server to generate all public/private keys made this option unsuitable for our system.

```
Miner Verification

For each transaction
        Connect (tx, ip)
        Send (random nonce)
Response ← receive
Verify (response, tx, nonce)
  ➔ Add to block mining queue ()
```

Figure 2: *Pseudo code for miner verification of a transaction*

The ability to prevent multiple identities from a single machine is key in preventing a Sybil attack, this combined with the expensive cost of entrance [25] to our network, makes it unviable for all but the most powerful adversary to conduct a Sybil attack on the network. To adapt this system for an E-commerce network, the data sent to the miners would be the Bitcoin transaction hash, the public key of the sender of the item and the public key of the receiver.

To reduce malicious transactions on the network, we also propose a proof-of-stake system, where a user with a low, or no, reputation stakes a small amount of currency (Bitcoins) into a triple signed wallet. A triple signed wallet is a wallet created with three sets of keys, one from the sender, one from the receiver and one from an impartial third party. When a low-reputation user wants to share a file, they demonstrate they are honest by sending a small amount of currency to the wallet set up especially for this transaction; this would mean if the user were to behave dishonestly and send a malicious file, the amount stored in the wallet would be sent to a pool which the network uses to act as a reward for miners finding blocks. This is chosen to discourage any user from trying to profit from this feature. If the transaction were to be conducted honestly, the file sender would receive the amount they staked back.

To ensure this network cannot be affected by a 51% attack in the early days of deployment we utilize the power of the Bitcoin network by using merge mining. Merge mining allows all miners on the Bitcoin network to use their hashing power on our reputation system. This does not reduce the hashing power of the Bitcoin network, but does increase the total hashing power of the reputation and thus the security of the reputation system, as now to conduct the 51% attack, an attacker would need to control the majority of computing power of both the Bitcoin and reputation network.

As well as the distributed blockchain, which ensures every peer has a full copy of the blockchain, eliminating client synchronisation issues as faced on previous distributed reputation systems, we also use the "friend peer reputation" model. As well as publishing all reputation about transactions onto the blockchain, the client also stores reputation from peers it has had previous interactions with. This can be multi-dimensional reputation, such as speed of the transaction, quality of file, etc. This information is not published to the blockchain as it would increase the cost of storage required per transaction and more importantly it is subjective from a user's perspective.

The final component of our reputation system is how to calculate reputation score of each peer. Reputation scores are not published on the blockchain. Unlike most previous generation reputation systems where the reputation client is community controlled, our proposed reputation system is client controlled. The client can calculate the reputation score based on parameters set by them. For example, a user could only view reputations from users on a specific network. To prevent against the collusion attack, where multiple users trade between themselves multiple times in order to unfairly gain reputation, each user will only be given a reputation score based on the average of all their reputation scores. This ensures if two nodes are transacting together, they will get the same reputation scores whether they send one transaction or a thousand transactions to each other.

For the network to have the property of temporal adaptability, the client could only rate users from reputation over a short period of time. Jøsang et al. [26] demonstrate a user's behaviour in the last few days is a more accurate indicator of the user's future behaviour than analysing all previous behaviour on the network.

To select a user they wish to download a file from, for example, a user finds all the peers which are hosting the file, the client then calculates the reputation for each peer using data from the blockchain and also using the friend peer reputation data to calculate a list of the most reputable peers. Only requiring the client to calculate reputation of a small subset of peers reduces the computational resources required by the client. Once the user has calculated the most reputable client they can initialize the download. This method of peer selection can be used for E-commerce and other type of networks.

V. LIMITATIONS, ANALYSIS AND SOLUTIONS

As with any network there are some limitations in the deployment and use of this network. The majority of the limitations we faced were due to fundamental flaws in the architecture of the blockchain protocol

Unlike the majority of peer-to-peer networks, where network growth is uncapped, and will continue to grow as long as new nodes join and stay in the network, a blockchain-based network has a hard limit on the number of transactions that can be processed per second. EBay currently processes an average of 23,148 reputation transactions per second, however due to requirement of a block being mined every ten minutes, and a maximum block size, our network would only be able to process 10 transactions per second. This is a significant reduction in the transactions our proposed network is able to process per second compared to a more traditional, previous-generation reputation system.

If the network were to receive more than 10 transactions per second, the miners would be forced to queue the reputation scores which would be included in a later block. This is not

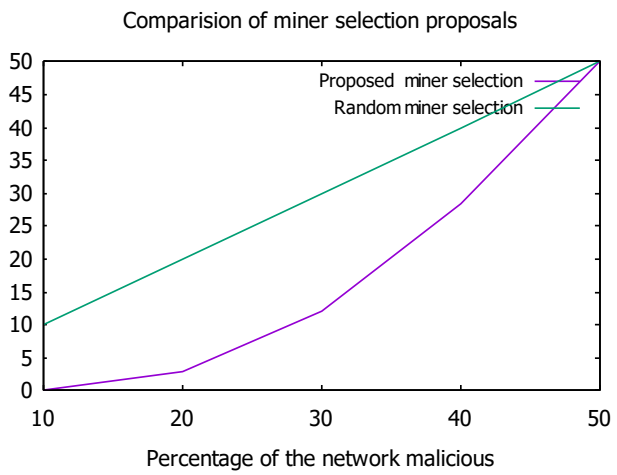

Figure 3: Comparison of miner selection algorithms

just an inconvenience to users who are relying on the network, it could also open the door for a denial of service where malicious colluding nodes would spam the miners with transactions, forcing miners to conduct computationally expensive verification of these transactions and forcing genuine users' transactions to be queued and delayed.

The "hard limit" on the number of transactions that can be processed a second also limits growth of the network and could render this application useless for some scenarios.

We will look at solutions to this problem later on in this section.

Another limitation on how effective and successful the reputation system is to be is the global deployment and adoption.

Currently, in addition to the issues mentioned above, another issue stopping this network from being deployed and implemented on a large scale is the resources required of each node. With the proposed 1MB block size - the same as the Bitcoin network - the blockchain could increase at a rate of 144MB a day (53GB a year). Needing to donate such a large resource in the network before being able to participate in it would be a significant barrier of entry to many users, especially low powered nodes, such as mobile users. This in turn would lead to growing centralization of a decentralized network.

These properties make it unlikely that a network with a high amount of low resourced users would implement this reputation system. This is a critical part of the success of the reputation system.

It would take several months from deployment for the reputation system to become effective, gaining the necessary data and feedback from users that would allow other users on the network to make informed decisions regarding the trustworthiness of a peer. It would therefore take several months from deployment before the full potential of this reputation system would be noticed.

While we have proposed a system that solves a number of known issues with current generation reputation systems, and which secures them using cryptographic functions, the risk of

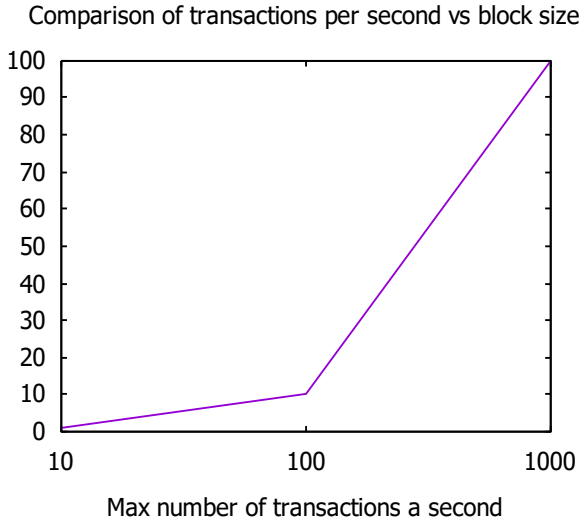

Figure 4: Graph showing relationship between block size and transactions a second

unknown technical flaws in the cryptography used could undermine security on the network.

The final limitation of the proposed network is un-defendable attacks, such as an intelligent colluding attack. While we have proposed countermeasures for such an attack, it might still be possible for an attacker to profit from the system. The impact of such an attack should be low, and with all the aforementioned countermeasures implemented such an attack would be very expensive to conduct, but we will never be able to defend against all possible attacks with 100% success rate.

VI. Analysis of Limitations

In the section we will analyze various methods of reducing the limitations of the network through simulations and calculations. We will also compare our proposed solutions to the current implementation and compare the results to other networks. One simple solution to increase the number of transactions per second would be to remove the maximum size of a block. This would increase the number of transactions per second the network would be able to compute, for example an increase to a 5MB block size would allow for 50 transactions per second. However, for this system to match eBay's 23,148 reputation transactions per second the block size would need to be 2.351GB, causing the blockchain to increase in size by 339GB a day; this is unsustainable and shows that increasing the block size is not the solution.

Fig 3 shows how the increased block size increases the number of transactions per second.

Another method to increase the transactions per second is to reduce the time required for each block to be mined. Currently the difficulty of the proof of work is calculated such that a block is confirmed every ten minutes. This could be reduced to 5 minutes, or even a single minute, to increase the transactions per second the network is able to process.

Both methods of increasing the block size would increase the resources required by the user, such as more storage space

to save the blockchain, as well as greater bandwidth to receive blocks at an increased rate. This would also further limit the participation of low-resourced nodes such as mobile devices. We therefore propose that each node is no longer required to download the entire blockchain, instead only the miners would be required to download and keep up to date the entire blockchain. This would change how reputations for users are calculated by the client; they would now be required to contact a pool of miners requesting the data for a specific user. A pool of miners will be used to prevent a malicious miner sending incorrect data to the requester, as in a pool, a majority of the miners would need to be malicious for this to occur.

We calculated the probability of randomly selecting a malicious pool (where 50%+ of the pool is malicious) for varying amounts of network compromisation, in comparison to randomly selecting a single miner using the equation below. We then simulated this model in Python before plotting the results on a graph as seen in Fig 4.

$$\rho(m) = \binom{k}{m} p^m (1-p)^{k-m}$$

As shown in Fig 4, this method is very effective for up to 40% of malicious nodes in the network, and effectively solves two limitations by allowing low-resourced users to join, as well as increasing the number of transactions per second. This demonstrates our proposed network is able to handle double the amount of malicious nodes supplying malicious reputation data as the reputation system proposed by Zhou [27].

To solve the blockchain size issue that is facing not only this network, but all blockchain-based networks, we have created a new method, which can reduce the blockchain size from the 53GB which Bitcoin's blockchain is currently, and is increasing, to a constant size of just 4.5GB, which would remain this constant size no matter the network size, or length of time the network has been deployed.

To achieve this, we are the first that have successfully implemented fully decentralized, and trustless checkpoints on a blockchain network, and to have created a self-deleting and self-managing blockchain.

Unlike previous solutions to the scalability issue, miners are not required to download the entire consensus, and delete "spent blocks" manually. Our solution aims to be globally agreed, and will no longer require any node to store the blockchain history from the first "genesis" block.

At a set point every day, our method requires the miners who are mining the current block to add a checkpoint to this block, in which all blocks older than 30 days can be safely removed. Since this occurs every 24 hours, only 24 hours of data (144mb) would be removed each day.

There are issues with this method, however, such as what if the miner implementing the checkpoint is malicious, and how can we prevent a user who has not received any reputation scores in the past 30 days from being deleted, as this would enable a modified re-entry attack to be conducted cheaply and easily.

Blockchain Roll

For the roll block:
←Receive(tx)
Validate(tx)
For each user to be delete(avg(rep), coinbase = coinbase+ avg(rep), delete(rep))
Hash of block(find nonce)
→Send(Block)

(x3)
← Receive (Block)
Validate (Block rx)
Validate tx(rx)
Hash of block + nonce
→Send(Block)

Delete (Block(data) from day(x))

Figure 5: Pseudo code showing the "roll" of the blockchain

To prevent a user being forgotten is perhaps the most important issue to solve. To do this, we exploit the structure of the blocks that can be inserted into the blockchain. The "genesis block" was able to initialize some accounts with a set value, this is known as the coin base, and is normally used to reward a miner 25BTC after they have completed the block race and confirmed a block.

We propose that, on this special block that would occur daily, the node would be required to search back through the 24 hours that would be due to be deleted from the blockchain and any users who would be completely removed from the block would have their reputation scores averaged and initialized into a new coinbase called "outdated users". By doing this, it would ensure no user would ever be deleted. On the next deletion period, if the user was still not active on the network, they would again be reinitialized in this special coinbase, however if they were active on the network, and not going to be removed from it, they would be removed from the "outdated users" coinbase.

Were a malicious node to be in control of the "roll", it could fake a reputation score for a specific user, or exclude select users from the network, effectively deleting their history from the network, something which we tried to prevent in the method described above. We do not consider the low probability of a malicious node being in this position to be strong enough security for this model.

To ensure this does not happen, nodes on the network would not just take this block as confirmation to delete previous data, instead, for the next 3 block, each node would check all of the work completed in the roll block is correct and honest. If it is, then the node signs this block and carries on as normal, if it is not then they do not sign the block, instead creating what they think is the correct block and signing that; this would cause the network to fork. After 3 blocks, the longest chain would be considered the valid chain and the other chain would be dropped from the network. The

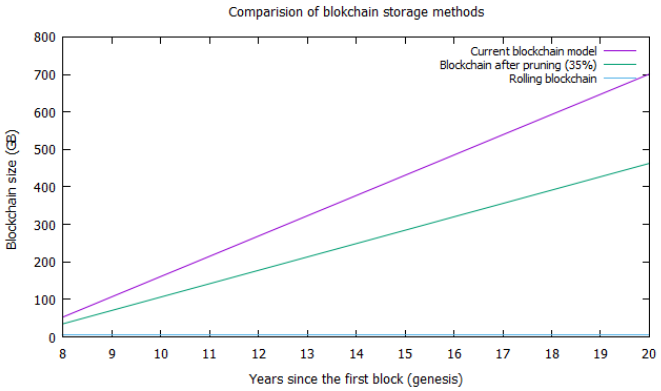

Figure 6: Comparison of blockchain scalability methods

probability of a malicious actor controlling the network to achieve four consecutive blocks without controlling a majority of the hashing power of the network is near zero. Once the "roll" block is confirmed, all nodes on the network would be able to delete all blocks older than 30 days from their memory, and thus reducing the amount of data they need to store to 4.5GB.

Fig 5 shows pseudo code of how the miners would conduct the deletion and how validation of this block is achieved following its confirmation.

Fig 6 shows how the blockchain would increase in size with time, from 2008 (first deployment of Bitcoin) comparing it with the Bitcoin blockchain with pruning enabled (35%) and also our proposed blockchain, deployed from the same date. (Assumption – all blocks will be of the maximum 1MB limit in size)

As shown in Fig 6, the blockchain grows constantly from years 8 to 20, dramatically increasing from 53GB in size to 701GB. The results start from year 8 of deployment because 2016 marks the 8th anniversary of Bitcoin's deployment, and it is easy to estimate the future, considering the majority of blocks today are 1MB. The pruning method, which reduces the size of the blockchain by 35%, but only after the user has downloaded the entire blockchain, increases from 35GB to 470GB at year 20, whereas the proposed model here stays at a constant 4.5GB, irrespective of how long it has been deployed.

The advantages of such a scheme is obvious, the blockchain size would be dramatically reduced to a constant fixed size, which would allow the participation of users who were previously excluded due to the cost of entry to the network.

We created a simulator to simulate this rolling blockchain; the blockchain was kept at a constant size of 4.5GB, and we found that the amount of network traffic caused by nodes joining the network was significantly lower than in traditional blockchain-based networks. The reason for this is that each node is now only required to download 4.5GB and not 53GB (or more), and on a network with an average of a 15.89% (1,038 nodes) turnover of nodes, this has led to a substantial reduction of network traffic of 50.34TB, daily. This reduction on network traffic would have an impact on reducing the propagation delay of transactions, which in turn would mean we would be able to increase the number of transactions able to be processed. It also reduces the cost of entry to the network, as with the reduced network traffic and download being required, users in areas with slow and limited internet connections are able to participate, something which on current blockchain-based networks has to date not been possible.

These improvements have made the cost of entry to the network 10 times cheaper in terms of resources than entry to traditional blockchain-based networks and the cost of participating in the network does not increase from the cost of entry, unlike previous generation blockchain networks.

This reduction in cost of entry and participation to the network could make the network vulnerable to attack, and particularly to the Sybil attack. It is important to conduct analysis to see if the tradeoff of cost of entry to the networks is an increased vulnerability of the network to attacks such as the Sybil attack; only a full deployment of the network would allow us to accurately assess if the reduce entry and participation cost does create a security risk to the network.

Another important security issue faced is the double spend attack, where a user can spend an amount twice, in our network we do not worry about this issue since a user cannot spend their reputation, however for this to be deployed on other blockchain based networks this was examined. It was found during analysis to be no more vulnerable to this type of attack than current generation blockchain networks and the cost of successfully conducting this attack was the same on both networks.

However, this model also has another advantage, if we assume all users have 50GB of memory to donate to the network (current size of the Bitcoin blockchain), we would be able to increase the transactions per second (TPS) from the current limit of 10 on blockchain-based networks to 100TPS on our network by increasing the block size to 10MB, which with current technology would not negatively impact the network. While this is still significantly lower than the current number of TPS on both eBay and VISA, it is a tenfold increase on the current blockchain-based networks without any additional memory resources being required.

A. Analysis of results

We have looked at the proposed reputation system and described some limitations faced during implementation. To ensure these limitations were mitigated, we developed a series of countermeasures to ensure the proposed network is as deployable as possible, in order for it to be successful.

The solutions to the limitation issues have now improved the scalability of the network. The countermeasures proposed and simulated in this paper could be implemented into any blockchain-based application which is having scalability issues.

Changing the block confirmation time from ten minutes to five not only aids with scalability of the network, doubling the number of transactions that can be processed per second, it also increases security, as now a malicious peer could be

detected 50% faster than before. This increase in detection time was an unexpected benefit.

There could however be negative impacts caused by our recommended changes to solve the limitation issues. The increased resources (storage space for the blockchain) on the miners could result in fewer miners on the network; this would in turn lower the security of the network, however the blockchain of the reputation system would still be significantly smaller than Bitcoin's blockchain for at least the first two years of deployment, so we do not see this actually happening. Another perceived negative impact is that the time for a peer to calculate a user's reputation will increase, this is due to the peer now needing to request this data from a pool of miners. The network latency and processing of this request would add a small delay, but this would not be significant enough for the user to notice.

The scalability of the network, and in fact all blockchain networks, was analyzed, with it being concluded that if the network were to be a success then scalability needed to be addressed. Proposed here is the first solution to the scalability issue, which aims to successfully delete data from the blockchain, and by doing so has also solved some fundamental issues regarding scalability.

We have successfully reduced the blockchain size by 92%, from 53GB to 4.5GB - a significant improvement over the current solution of pruning, which offers just a 34% reduction from 53GB to 35GB. In addition, we are the first to show how trustless decentralized checkpoints can be implemented on the network.

The proposed method has shown no loss in security compared to the standard blockchain model, but has been shown to significantly reduce the network traffic by 50.34TB, with the result that users who have previously been excluded from fully participating in blockchain-based networks are now able to do so, which in turn both increases the network's security and allows the network to have more resources at its disposal.

The lower cost of entry has theoretically made the network more secure, as every node which participants in the network also secures it by donating computing power to securing the blockchain. This means attacks such as the 51% attack would incur increased deployment costs. It is impossible to estimate the increase in security, as there are no statistics regarding low resourced users being excluded from the previous generation of blockchain networks. So to test this theory and conduct further analysis, we would need to deploy the network live.

VII. Conclusion

In this paper we have discussed a next generation reputation system based on the blockchain, we have shown how a generalized reputation system that could be implemented into various networks is possible. We discuss in detail how the reputation system would be implemented and demonstrate how our proposed system solves many of the issues faced by current reputation systems. We conducted analysis on the limitations faced by our system before describing how these could be overcome, and have proposed a solution to the scalability issue, which not only affects this network, but every network based upon the blockchain.

Overall, this paper aimed to propose a reputation system which solves the majority of issues faced in current reputation systems. However, this is just the foundation of the idea and there is a lot more research to be conducted in the future in various areas to ensure this reputation system is capable of replacing all reputation systems in the real world.

VIII. Future work

This paper has shown how a reputation system could be easily implemented on a blockchain, and how our proposed reputation system theoretically solves the majority of issues faced by current generation systems. However, this is just the beginning of development of this network, and there are still many avenues of research left to pursue in this area.

The most important piece of work to conduct in the future is to make this proposed network live. This will then let us examine in greater detail if the assumptions in this paper hold true in the real world.

We cannot yet answer questions such as whether a user who acts honestly on one network can be assumed to act honestly on all networks they interact with, or when past reputation for a user becomes irrelevant, but with more research we hope to be able to resolve these questions and more besides.

Deployment onto a live network would also enable more accurate analysis of how users interact with the reputation system to allow a more accurate algorithm for calculating reputation scores to be refined.

Deployment onto a real world network would also allow us to see if our solutions to known issues and limitations hold true, or if new issues surface.

This paper has so far assumed a user does not worry about privacy, however there is a growing consensus that privacy is a critical factor in using any web application, so it would be a very interesting research area to consider if privacy can be implemented on a reputation system without succumbing to attacks which exploit the weak links between identity and users.

We have focused on two applications for this system; an Ecommerce eBay type application where users can rate if they received the item, and also a peer-to-peer network, where users can rate each other peer if they have provided the correct file, in an attempt to detect any malicious nodes spreading malicious files through the network. It would be beneficial to the future success of this network if other implementations in these applications where possible. For example, instead of just rating a peer on whether it sent the correct file in a peer-to-peer network, could this system be adapted to bittorent and used to provide each client with the optimum download and upload speed, allowing each users to rate a series of other criteria to provide a better service to the client.

An interesting area of research which is being continued is the question of scalability of blockchain-based networks. In this paper, we proposed and simulated one solution to this

issue, however more research is required to ensure that ours is the best solution for this problem, and to be able to implement this solution on a live network to accurately measure the results for comparison to our simulator.

The final area for future research is how to optimize the blockchain. Could pruning the blockchain be a possibility in this situation, as this would allow the network to scale higher due to the lower resources needed.

These are just some of the interesting research areas that we have yet to fully analyze, and with more research this project could be the next generation of reputation systems.

REFERENCES

[1] G. Prisco. (2015, May 14).*The World Table Launches a Quantified Reputation System* [online]. Available: https://bitcoinmagazine.com/articles/world-table-launches-quantified-reputation-system-1431633676

[2] WSO2. (2011). *EBay uses 100% Open Source WSO2 Enterprise Service Bus to Process more than 1 Billion Transactions per Day* [online]. Available: http://wso2.com/download/wso2-ebay-case-study.pdf

[3] Ebay. (2015). *Changes to Feedback – FAQ* [online]. Available: http://pages.ebay.co.uk/help/sell/feedback-faq.html

[4] P. Dewan and P. Dasgupta, "Securing reputation data in peer-to-peer networks", in Proc. of Parallel and Distributed Computing and Systems (PDCS 2004), Cambridge, MA, 2004.

[5] M. Gupta, P. Judge and M. Ammar. (n.d.). *A Reputation System for Peer-to-Peer Networks* [online]. Available: https://www.cs.indiana.edu/~minaxi/pubs/reputation.pdf

[6] S. Schiffner, S. Clauß and S. Steinbrecher. (2007). *Privacy, Liveliness and Fairness for Reputation* [Online]. Available: https://securewww.esat.kuleuven.be/cosic/publications/article-1421.pdf

[7] Y. Wang and J. Vassileva, "Trust and Reputation Model in Peer-to-Peer Networks," in *Proc. of the 3rd Int.Conf. on Peer-to-Peer Computing*, Linköping, Sweden, 2003, pp. 150-157.

[8] S. Kamvar, M. Schlosser and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks," in *Proc. of the 12th Int. Conf. on World Wide Web*, New York, NY, 2003, pp. 640-651.

[9] P. Chewelos and T. Dhar. (2009). *Differences in "Truthiness" across Online Reputation Mechanisms* [Online]. Available: http://www.sauder.ubc.ca/News/2007/~/media/Files/Faculty%20Research/Chwelos-Dhar-OnlineTruthiness.ashx

[10] K. Hoffman, D. Zage and C. Nita-Rotaru. (2007). *A Survey Of Attacks On Reputation Systems* [Online]. Available: http://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=2676&context=cstech

[11] A. Jøsang and J. Golbeck, "Challenge for Robust Trust and Reputation Systems," in Proc. of 5th International Workshop on Security and Trust Management, [2009] © Elsevier Science B.V, Sept. 2009

[12] G. Greenwald. (2014, February 24). *How covert agents infiltrate the internet to manipulate, deceive, and destroy reputations* [online]. Available: https://theintercept.com/2014/02/24/jtrig-manipulation/

[13] J. Douceur. "The Sybil Attack," In *IPTPS '01 Revised Papers from the First International Workshop on Peer-to-Peer Systems*, Cambridge, MA, 2002, pp. 251-260

[14] G. Danezis and P. Mittal, "SybilInfer: Detecting Sybil Nodes using Social Networks," in *Proc. of the 16th Annu. Network and Distributed System Security Symp.*, San Diego, CA, Feb 2009, n.p,

[15] H. Yu et al, "SybilLimit: A Near-Optimal Social Network Defense against Sybil Attacks," in *2008 IEEE Symposium on Security and Privacy*, [2008], © IEEE. doi: 10.1109/SP.2008.13

[16] B. Prêtre, "Attack on Peer-to-Peer Networks," Semester Thesis, Dept. of Computer Science, Swiss Federal Institute of Technology (ETH) Zurich, 2005

[17] K. Aberer and Z. Despotovic. (2001). *Managing Trust in a Peer-2-Peer Information System* [Online]. Available: http://lsir.epfl.ch/aberer/files/PAPERS/CIKM2001.pdf

[18] J. Poon and T. Dryja. (2016). *The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments* [Online]. Available: https://lightning.network/lightning-network-paper.pdf

[19] Visa. (2015). *Visa Inc. at a Glance* [Online]. Available: https://usa.visa.com/dam/VCOM/download/corporate/media/visa-fact-sheet-Jun2015.pdf

[20] Hashing It. (2014). *7 Transactions Per Second? Really?*[Online]. Available: http://hashingit.com/analysis/33-7-transactions-per-second

[21] I. Eyal, A. Gencer, E. Sirer and R. van Renesse. (2015). *Bitcoin-NG: A Scalable Blockchain Protocol*[Online]. Available: http://arxiv.org/abs/1510.02037

[22] S. Nakamoto. (n.d.). *Bitcoin: A Peer-to-Peer Electronic Cash System [online].* Available: https://bitcoin.org/bitcoin.pdf

[23] Blockchain. (2015, October). *Hash Rate* [online]. Available: https://blockchain.info/charts/hash-rate

[24] R. Cohen. (2013, November 28). *Global Bitcoin Computing Power Now 256 Times Faster Than Top 500 Supercomputers, Combined!* [online]. Available: http://www.forbes.com/sites/reuvencohen/2013/11/28/global-bitcoin-computing-power-now-256-times-faster-than-top-500-supercomputers-combined/

[25] A. Mohaisen and J. Kim, "The Sybil Attacks and Defenses: A Survey," *Smart Computing Review*, vol. 3, no. 6, pp. 480-489, Dec. 2013

[26] A. Jøsang et al. (n.d.). *Simulating the Effect of Reputation Systems on e-Markets* [online]. Available: http://folk.uio.no/josang/papers/JHF2003-iTrust.pdf

[27] R. Zhou and K. Hwang, "PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 4, pp. 460-473, Apr. 2007