2019 IEEE Conference on Communications and Network Security (CNS): Workshops: DLoT: 2nd International Workshop on Distributed Ledger of Things

Distributed Ledger for Spammers' Resume

Anudeep Sai Muttavarapu
Department of Computer Science and Engineering University of North Texas
Denton, TX
anudeepsaimuttavarapu@my.unt.edu

Ram Dantu
Department of Computer Science and Engineering University of North Texas
Denton, TX
ram.dantu@my.unt.edu

Mark Thompson
Department of Computer Science and Engineering University of North Texas
Denton, TX
mark.thompson2@unt.edu

Abstract—**Unsolicited, and most likely spoofed, robot calls are not just an annoyance, but also carry a potential threat with the onset of automation, impersonation, and even voice manipulation technologies as malicious elements attempt to use deception to steal sensitive information or invoke action. Despite steps taken to protect consumers, the issue appears to be far from under control. In this paper, we propose a solution to use blockchain as a platform to share spam transactions through a peer-to-peer mechanism that will maintain a global database of reported spam transactions in order to identify and trace spam activity effectively. Storing spam transactions on a distributed ledger with consensus-based approval of transactions adds reliability to the data and can optimize the data points that will be available to spam detection algorithms in order to fight spam effectively. As this is peer to peer-based sharing, there is no need to rely on third-party providers for storing and sharing this data to the users. Every spam call received will be added as a detailed transaction on the blockchain to execute a smart contract that will calculate the trustworthiness of the caller. Call records are used to identify spam transactions while the blockchain ledgers store this data. We discuss the relevance and advantages of a distributed ledger to store these transactions. This paper does not aim at solving the spam problem with an optimized detection algorithm but evaluates the characteristics and performance of the blockchain as a distributed ledger and its relevance to serve as a platform for peer-to-peer spam detection mechanisms. We evaluate different blockchain metrics like transaction processing rates, gas costs and ledger sizes and discuss how they scale in order to store the spam reports data on the blockchain.**

Keywords- Spam detection, Blockchain, Robocalls, peer-to-peer, Smart Contract.

I. Introduction

A robocall [3] is a phone call that uses a computerized automatic dialer to deliver a pre-recorded message, as if from a robot. Spammers make use of robocalls to dial users with a prerecorded message or an interactive voice response mechanism to deliver spam content to users. Auto-dialing software that incorporates Voice over Internet Protocol (VoIP) [1] calling technologies are typically used to make these calls on a large scale. Caller IDs are usually spoofed by these autodialers to make the victims believe that the call is made by a local caller or organization. These robocalls are not only used to deliver spam content to the users, but also to trick users into believing that the call is from a government organization such as the IRS or FBI to steal information such as credit card and Social Security numbers.

According to YouMail Inc [2], approximately 30.5 billion robot calls were received by American consumers and businesses in 2017. These numbers show a 19.2% increase in these calls over the previous year with this trend expecting to continue. In the month of December 2017 alone, an astonishing 89.6 million robot calls were received by consumers. On average, 967 of these type of calls are placed every second.

A detailed survey done by Huahong Tu, Adam Doupé, Ziming Zhao, and Gail-Joon Ahn [4] at ASU outline possible technologies and solutions to tackle and prevent robot calls that are available today. The survey also identifies specific advantages and drawbacks of each of these technologies and concludes that there is no universally acceptable solution due to various factors like usability, deployability, and robustness. Most of the available solutions lack the type of integration needed to coordinate among the various participants of this ecosystem. Existing solutions include carrier-specific ways to fight spam such as AT&T Call Protect [5] and T-Mobile Scam Block Scam ID [6].

Victims can also use third-party apps such as True Caller [7] and Robo killer [8] to detect and block robocalls. Device OS developers like Google are providing technologies in Android to combat these robocalls [9]. As another option, a user can register for the Do Not Call Registry [10] to stop receiving these calls; this is unfortunately not always effective. Though there are many novel methods that are available to detect spam, these cannot be effectively implemented unless the different types of spam detection mechanisms from the above-mentioned sources can be collaboratively used on a common platform with transparency and auditability.

In this paper, we propose to use the blockchain platform to serve the purpose of peer-to-peer spam sharing and user reputation tracking. This architecture provides a decentralized way of storing and tracking spam records that can be used to derive user reputation. It also guarantees transparency and auditability of transactions submitted by users. With such a decentralized way of sharing spam transactions, users can now adopt different types of spam detection mechanisms while still

978-1-5386-7117-7/19/$31.00 ©2019 IEEE

being able to update a global ledger, thus making it more robust and reliable.

A. Why Blockchain

Existing spam detection applications such as true caller is based on centralized platforms and usually rely on one method or algorithm to detect spam. Also, Users do not have visibility about how the data is used to arrive at a conclusion if a phone number is spammer or not. There is a need for transparency of transactions so that users can know the series of events that affects the reputation of the phone number. The integrity of such data must be preserved, and any third-party provider should not be able to change the data. Users should be able to have a trace of spam transactions and the state of the reputation at any moment and the series of events that led to the current reputation score of a user.

By using blockchain and providing a distributed application that any user can query and update, we have achieved below advantages over the traditional centralized mechanisms. These advantages are achieved due to the inherent properties of blockchain without making any explicit changes to how blockchains work.

1) Distributed: The spam reports submitted by a user are stored over a distributed ledger. Blockchain is a shared repository that is maintained by peers — everyone can access data and view transactions. This is necessary so that the reputation and the series of spam transactions can be openly viewed by users at any given point of time.Moreover, storing information on multiple nodes prevents data loss in case of unexpected events. Multiple spam detection techniques can be used to update the ledger, which can be used to calculate phone number ratings. It has the properties to run without a central authority and cannot be censored. Existing spam detection applications does not provide these features.

2) Transparency: Users trust is very important in reputation-based spam detection mechanisms. In blockchain, the spam transactions reported and stored on the blockchain can be accessed and reviewed by all the peers. All the participants can read not only the final state of transactions, but also the history of past states. This data present in the distributed ledger can be trusted by users as all the transactions are publicly visible and users can query the ledger to obtain the data at any point of time. This will build trust among the users and users will more actively participate in the spam sharing mechanism and they can reply on results from such a mechanism which is lacking in existing applications.

3) Auditability: Each and every spam transaction reported can be audited by users before it becomes part of the ledger. All the users can read the history of past states as well as the final state of the transactions. The data is only persisted after the inherent mechanism of validation using mechanisms such as proof of work, so the integrity of data present on the ledger is preserved.

4) Immutability: Unlike the data stored on centralized databases by third party providers, data can never be erased or modified once committed.

A global ledger that is trusted by all users with a trace of spam transactions and reputation history available for audit at any point of time will gain users trust and users will more actively participate in reporting spam transactions. This also means users can choose different spam detection mechanisms off chain like caller behavior analysis and speech processing, but they will still be able to update a global ledger.

This paper evaluates the technical performance to validate whether blockchain can be used in peer-to-peer spam detection to create and share spam reports among the users. Each user can have his own way to do this spam detection off-chain, for example, in caller behavior analysis and speech processing, which will update the global ledger that calculates the rating of the phone number. All the peers can query the blockchain to acquire the phone number rating and hence know if the caller is a spammer.

The rest of the paper is structured as follows. Section II describes the system architecture and defines the details of the ledger, smart contract, and various transaction flows. Section III defines the details of our experimental setup environment that supports the architecture. Section IV contains the performance evaluation of the blockchain for the experimental setup while Section V concludes the paper.

II. System Architecture

In this paper, we discuss the relevance of blockchain to serve as a platform for peer-to-peer spam reporting and sharing using a reputation-based system that leverages blockchain to store the transactions, thus inheriting valuable blockchain properties such as auditability, trust, and distribution nature. Our implementation is composed of six distinct components.

A. Participants in the Blockchain

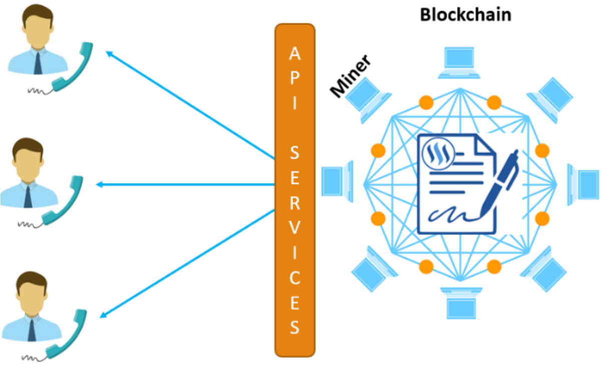

Fig. 1. Participants in the blockchain

- Users: Users of a telecommunication service can participate collaboratively to report spam transactions onto the chain and assist others to avoid spam calls.
- Miners: These are nodes that process and approve the transactions using PoW/PoA. Miners keep adding new blocks to the chain as they approve more and more transactions. Miners serve as the sources for a query to the trust rating of a given user upon reception of a call.
- API Services: These are nodes that offer REST API's for users to query the rating of a given caller. The APIs can also be used to report spam onto the blockchain. The API services node contacts a miner to update the blockchain with spam transactions or query the rating of a given caller.

B. Ledger Structure

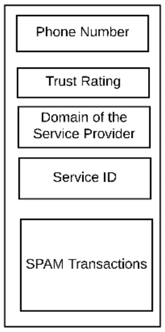

Fig. 2. Ledger structure

The ledger is composed of the following data:

- Phone number of the user
- Trust rating of the phone number
- Spam transactions
- Domain of phone number
- A unique ID on the blockchain

C. Smart Contract

A smart contract hosted on the blockchain will support the following functionality:

- phone number registration and creation of a digital identity for the service based on the service ID,
- an interface for users to report spam calls, and
- update of the trust rating of the phone number based on the spam reports.

The specific data structures utilized in the smart contract are outlined in Fig. 4.

The register phone number structure stores the username , an unique identifier validated from oracle and a Service ID assigned at the time of registration and a registration flag that is used to identify if the user has voluntarily registered or he is registered automatically due to a SPAM transaction reported. The create service structure contains a mapping from username to the properties like phone number, trust rating and the reference to list of spam reports. The phone number is also mapped to a domain which has details like domain name and domain rating. Each spam report is stored which is referenced by the spam reports hash.

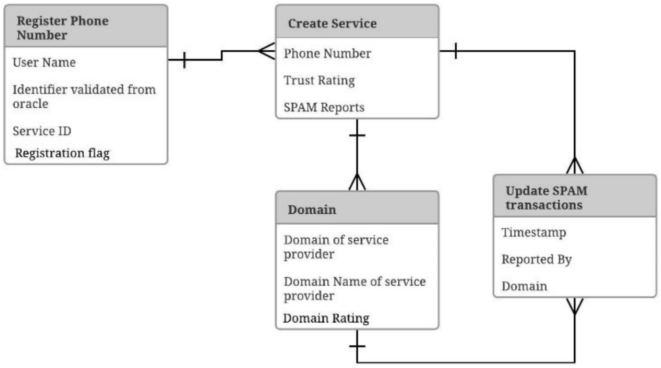

Fig. 4. Smart contract data structures

D. Supported Flows

Below is the flows supported on the application:

- Users can register on the blockchain with their phone number.
- Users can report a spam call to the blockchain in real-time.
- The blockchain stores the spam transactions while smart contracts calculate the trust rating of a given service ID.
- Spam call transactions reported by the user will be stored on the blockchain with details including caller ID, timestamp, etc.
- Internet telephony service providers can track the on-chain records and match them with their local logs to trace back to the originator of the spam and take appropriate action.
- Real-time APIs will be available to query the trust rating of a mobile number, which can be used to alert users when a call is received with that caller ID.

E. REST APIs

Fig. 6. Shows the REST API's [13] that are available for the user to interact with the smart contract. The REST API's are categorized as follows:

- Registration API's: These API's are used to register phone numbers and their respective domain on the blockchain ledger.

- Reporting API's: These API's are used to report spam and update the trust rating of the phone number.
- Access API's: These API's can be used to read phone number ratings, domain ratings, and basic phone number details.

REST API's are introduced since some existing mobile clients may not be capable of interacting with the blockchain directly using web3.js. REST API's also ensure that cross-platform interaction with the blockchain can be facilitated. These API services act as a medium to communicate with the blockchain and do not directly change any data. They are just pass-throughs that convert calls from HTTP to RPC calls on the blockchain using web3.js.

Manufacturers like HTC [14], however, are introducing phones [15] that can interact directly with the blockchain. Once such devices are made widely available, REST API's can be eliminated.

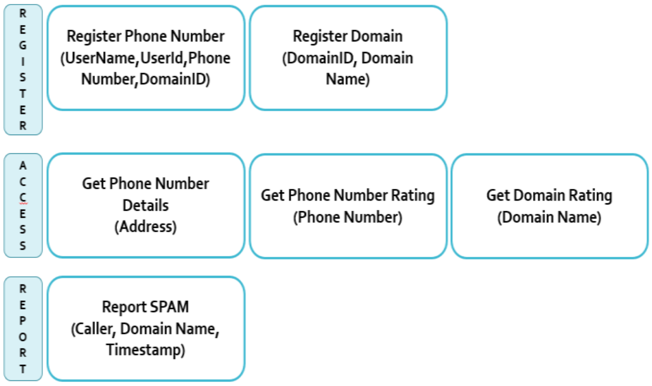
REGISTER
Register Phone Number (UserName,UserId,Phone Number,DomainID)
Register Domain (DomainID, Domain Name)
ACCESS
Get Phone Number Details (Address)
Get Phone Number Rating (Phone Number)
Get Domain Rating (Domain Name)
REPORT
Report SPAM (Caller, Domain Name, Timestamp)

Fig. 5. APIs exposed to interact with smart contract

F. Smart Contract Flows

The business logic flows inside the smart contract are detailed in Fig. 7 and Fig. 8.

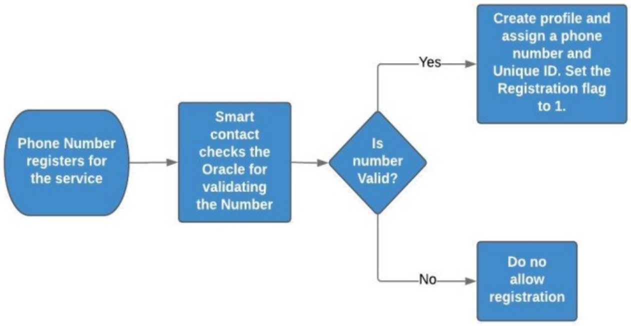
Phone Number registers for the service
Smart contact checks the Oracle for validating the Number
Is number Valid?
Yes
Create profile and assign a phone number and Unique ID. Set the Registration flag to 1.
No
Do no allow registration

Fig. 6. Register phone number flow

When a phone number registers on the blockchain using the Registration API, the smart contract verifies with an Oracle to make sure that the phone number is valid. An oracle, in the context of blockchains and smart contracts, is an agent that finds and verifies real-world occurrences and submits this information to a blockchain to be used by smart contracts. This can be a mechanism such as One Time Password (OTP) [16] that validates that the user owns the phone number. As this paper concentrates on evaluating the blockchain performance, OTP is not implemented in the current version described in this paper. In real-time implementations, however, validations like OTP can be implemented to make sure the user is indeed the owner of the number. Once the phone number is validated and registration succeeds, the phone number will be added to the ledger.

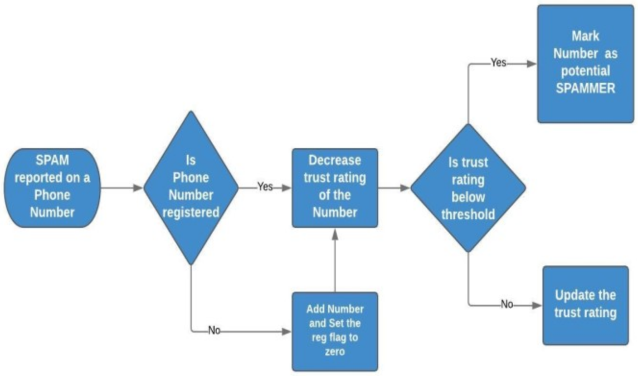
SPAM reported on a Phone Number
Is Phone Number registered
Yes
Decrease trust rating of the Number
Is trust rating below threshold
Yes
Mark Number as potential SPAMMER
No
Update the trust rating
No
Add Number and Set the reg flag to zero

Fig. 7. Report spam flow

When spam is reported, the smart contract checks if the phone number is registered on the blockchain. If the phone number is not registered, the smart contract adds the number to the ledger and sets the registration flag to 0, indicating that the phone number has not been validated, but instead automatically added since a spam transaction has been reported against the phone number. The smart contract then continues to update the trust rating of the phone number and marks the phone number as a potential spammer if the rating falls below the specified threshold.

III. EXPERIMENTAL SETUP

Below is the experimental setup that is used for the project.

i) Node.js, ii) Express, iii) Web3.js, iv) Ropsten test net/ Ethereum mainnet, iv) Remix Web IDE, and v) Truffle.js.

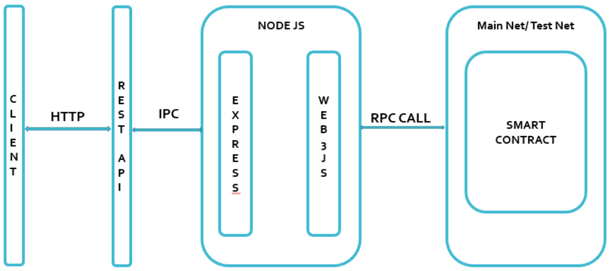

Fig. 8. Architecture Overview

1. Users can use HTTP (REST client) to query the blockchain for trust ratings and to report spam transactions onto the blockchain.
2. API services are built using Node.js with a Web3.js client to interact with the blockchain. The service will find the nearest blockchain miner and connect to the blockchain to post the spam transactions and query the phone number ratings.
3. Ganache is used as the blockchain during the development phase of the smart contract. Performance results, however, are measured on the Ropsten test net and Ethereum mainnet.
4. The Truffle HD Wallet provider plug-in is used to interact with the blockchain for user accounts and sign transactions.

A. *Flows*

1) *Report spam:*

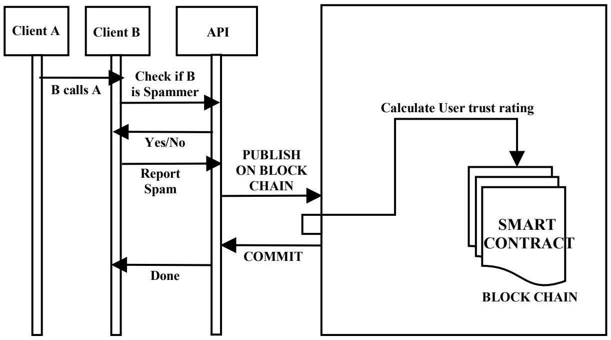

Fig. 9. Report spam flow

a) *B* calls *A*, where *B* is a spammer.

b) *A* reports a spam transaction to the blockchain using the REST API so that the transaction will subsequently be posted to the blockchain.

c) Other users on the network then retrieve all the older spam reports on *B* and validate the transaction.

d) Once the transaction is validated, it is saved on the blockchain.

e) The smart contract recalculates the trust rating of *B* and reduces the trust rating.

f) Users can query the blockchain for the trust rating of a phone number.

2) *Register Phone Numbers:*

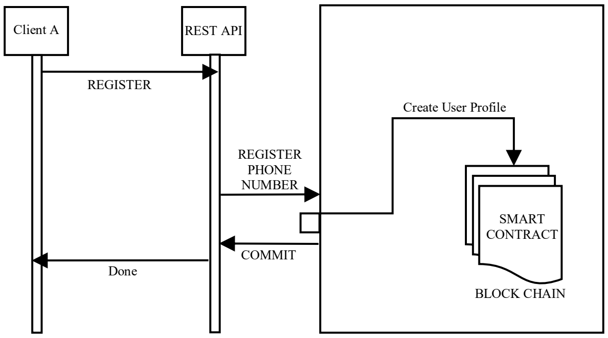

Fig. 10. Register phone number flow

a) Phone number registers with the service.

b) A profile is created on blockchain.

IV. Performance Evaluation

Testing shall be done in both the Ropsten test net and Ethereum mainnet. Test data is generated based on metrics that measure the relevance of blockchain to serve as a platform for peer-to-peer spam detection are measured and characterized as follows:

A. *Metrics in the analysis:*

1. Transaction receipt times and their variance on the Ropsten test net and Ethereum mainnet:

This performance measurement is done to estimate the time its takes to run these transactions on block chain and compare the transaction receipt times in Ropsten test net and Ethereum mainnet. An average of time taken from submitting the transaction to getting the receipt over 10 transaction attempts for each of the function is listed below:

TABLE I.
Transaction times on Ropsten test net

Function	Average Transaction Time	Minimum Transaction Time	Maximum transaction time	Gas Price (Gwei)
Register Domain	32.40 Seconds	4 Seconds	89.5 Seconds	1
Register	54.74 Seconds	4.5 Seconds	117 Seconds	1

Phone Number				
Report Spam	33.29 Seconds	3.17 Seconds	118.7 Seconds	1

TABLE II.
Transaction times on Ethereum mainnet

Function	Average Transaction Time	Minimum Transaction Time	Maximum transaction time	Gas Price (Gwei)
Register Domain	596.66 Seconds	395 Seconds	1113 Seconds	5
Register Phone number	581.22 Seconds	320 Seconds	890 Seconds	5
Report Spam	613.33 Seconds	343 Seconds	1223 Seconds	5

Observations:

The average transaction receipt times on the Ropsten test net ranged between 30 - 55 seconds for 10 different transactions. In Ethereum mainnet, the average transaction receipt times were significantly higher than the Ropsten test net with the values ranging between 581 - 613 seconds for the various transactions. On Ethereum mainnet, the gas price paid for the transactions was 5 gwei whereas on the Ropsten test net it was 1 gwei. The gas price on mainnet was derived based on Eth gas station [24], which provides the recommended gas price based on the current network status of the Ethereum.

The transaction times are dependent on two main factors: the number of pending transactions in the transaction pool and the block creation time. Ropsten test net did not have a large amount of pending transactions during the time test transactions were run. This information is validated based on Ropsten Block Explorer [25]. However, Ethereum mainnet had approximately 36,000 pending transactions at the time of testing based on etherscan [17]. The miners compete to pick transactions that offer higher gas price over the ones with lower gas. Hence a higher transaction receipt time is found on Ethereum mainnet whereas the transaction receipt times on Ropsten test net were shorter.

2. Variance of transaction receipt time with gas price:

In order to achieve faster transaction receipt times, a higher gas price is offered, where the transaction times are measured as the gas price is increased in Fig. 12. Ethereum blockchain has a transaction pool into which all the transactions are queued once they are submitted. Miners pick the transactions with the higher gas price with high priority, execute them, and create a block.

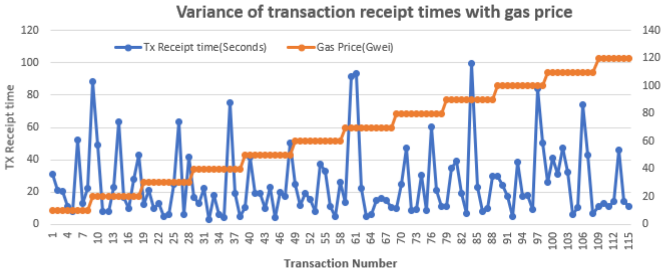

Fig.11 Variance of transaction receipt times in Ropsten test net

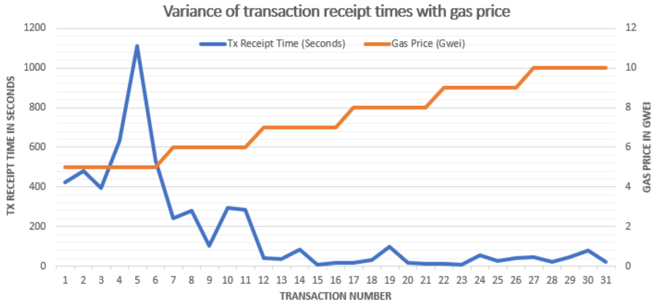

Fig.12 Variance of transaction receipt times in Ethereum mainnet

Observations:

In the Ropsten test net, increasing the gas price had no impact on the transaction times as there were no pending transactions for which miners were competing to solve. In Ethereum mainnet, however, increasing the gas price offered had an impact on the transaction times. The transaction times decreased when the gas offered was increased. The miners automatically gave high priority to the transactions that offered to pay a higher gas price, resulting in the transaction receipt times being reduced.

3. Time to read the phone number ratings:

The time to read the phone number ratings from the blockchain as the number of phone number and spam transaction increase is given below, where the *x*-axis represents the number of transactions while the *y*-axis represents the time in seconds.

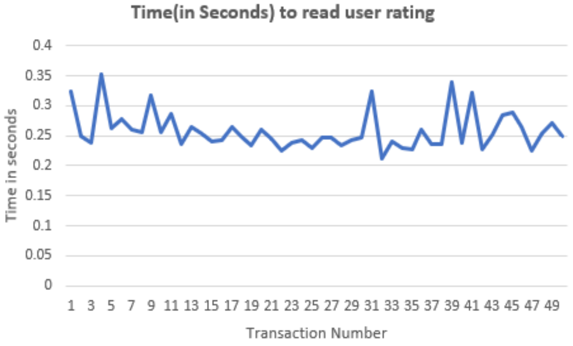

Fig.13. Time to read the phone number ratings on Ropsten test net

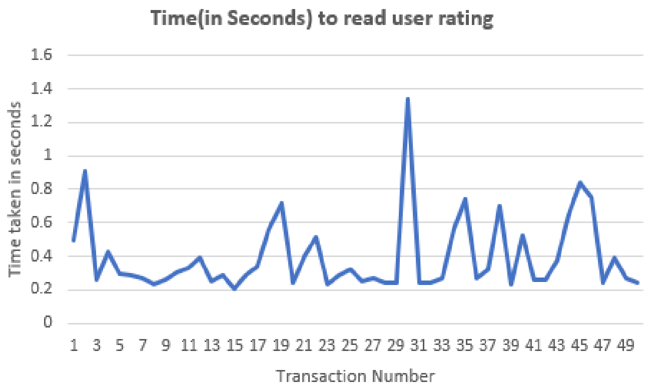

Fig.14 Time to read the phone number ratings on Ethereum mainnet

Observations:

Gas cost is zero as these are view only functions. As such, Ether is not required to view the transactions. A high volume of reads is expected, so this is a good advantage. The transaction times remain in milliseconds and provide real-time reads on phone number and domain ratings. This is an advantage of the blockchain as this is required to be in real-time.

The ledger access times are slightly higher in Ethereum mainnet compared with Ropsten test net.

4. Amount of gas spent on the smart contract execution:

Below are the gas costs for the transactions

TABLE III.

Gas costs for smart contract functions in Ropsten test net and Ethereum mainnet

Function	Gas cost in ropsten test net	Gas cost in Ethereum main net
Deploy smart contract	994905	1023029
Register Domain	69161	69232
Register phone number	129814	130019
Report Spam	165046	165799

Observations:

Gas spent for the transactions remained constant throughout the simulation. The gas is directly dependent on the size of the variables passed as inputs to solidity function. Gas consumed was constant even when the execution was tried on different times of the day/week. Gas consumed increased if the size of the variables passed was increased.

The gas spent is slightly higher on the Ethereum mainnet than on the Ropsten test net.

5. Estimation of costs of running the transactions in Ethereum main net:

This section compares the various metrics on the Ethereum network with application specific metrics to arrive at a cost estimate for this system per day. From [2], it is seen that users in the United States face around 967 spam calls every second. This means that there are 83,548,800 spam calls made daily. Assuming that each of these spam calls is reported by users, below will be the cost estimate for the system:

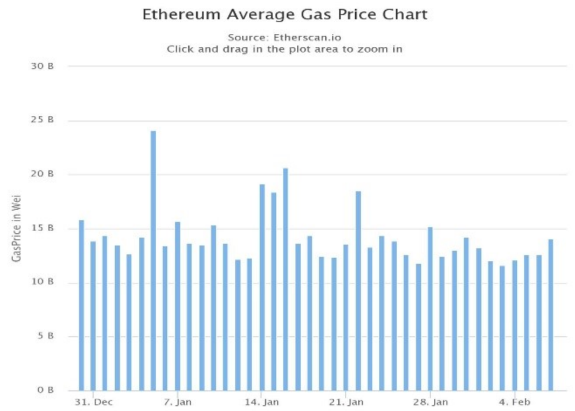

Fig.15 Ethereum average gas price chart [26]

Fig. 16 shows the average gas price in gwei in the month of January 2019. The max value was 24.12 gwei on the 5th of January while the minimum was 11.78 gwei on the 27th of January. The price of reporting spam at this rate would be ((amount of gas * gas price) * number of transactions). Based on the data, it can vary between 162,439.07 to 332,600.21 Ether. At the time of formulating this paper, the USD conversion of the same is ranging between $22,388,977.02 to $45,842,286.94 per day. This calculation is based on each spam call being reported as a transaction. Optimizations can be made, however, to report spam calls at periodic intervals, which will reduce the overall number of transactions.

Ethereum is an expensive platform for an application that requires a very high number of transactions since each transaction involves spending Ether to incentivize miners for the proof of work. Hence, the number of transactions must be effectively reduced by increasing the reporting interval of the transactions as well as summarizing and reporting the spam transactions in less frequent intervals.

V. Conclusions

Blockchain provides added functionalities to the existing spam detection techniques and promotes a peer-to-peer spam sharing mechanism with decentralization, auditability, and trust. Users will be able to share and update trust ratings without the involvement of third-party service providers. The data can be further used to derive metrics like spam call patterns geographically and can enhance the methods used to track and locate spam activity.

Below is a discussion on various conclusions on the usage of blockchain as a distributed ledger for storing and sharing spam transactions:

A. Transaction Approval Times:

The transaction approval times when a spam call is reported is a variant that cannot be determined since it directly depends on the miner activity and several other transactions running on the blockchain. Acceptable averages of approximately 33 seconds were achieved on the Ropsten test net and 613 seconds on Ethereum mainnet. This can be overcome by increasing the gas price. However, this is not a limitation as this need not be in real-time.

B. Concurrency of transactions :

Another requirement that needs to be satisfied is the support of concurrent transactions. From [2], in the month of December 2017, an astonishing number of 89.6 million robot calls were received by consumers. On average, 967 calls are placed every second. For blockchain to serve as a preferred platform for peer-to-peer spam sharing, it needs to support around 1000 transactions per second (tx/sec), but this number keeps increasing.

Here is an evaluation of how the different blockchain platforms work for the above requirement:

1. Ethereum:

According to Ethereum Transaction growth chart [17], the highest number of the 1,349,890 transactions occurred on Thursday, January 4, 2018, achieving a transaction rate of 15.62 tx/sec.

Theoretically, with a block gas limit of 801,111 [18] with the gas cost around 21,000 for each transaction, we achieve approximately 380 transactions per block. With the current block time of 15.03 seconds [19], Ethereum can theoretically support 25.346 tx/sec. This number is obviously below what is needed for real-time support for spam reporting.

However, Ethereum is actively working to bring this number up by different techniques as mentioned in [20] as follows:

- improving the sharding techniques by methods such as super-quadratic or exponential sharding [20] and
- improving the consensus mechanism by replacing proof of work with PoS beacon chain using Casper FFG for finality.

These techniques can bring down the block time drastically, hence increasing the tx/sec.

Daniela Mechkaroska et al. discuss various ways to increase the scalability of the blockchain [23]. Christian Decker et al. [27] suggest optimizations in the blockchain protocol in order to reduce the block propagation delays.

We are optimistic that these type of optimizations and changes can help Ethereum to be able to serve the transaction rate required for peer-to-peer spam detection.

2. Hyperledger:

Hyperledger [22] is an open source collaborative effort created to advance cross-industry blockchain technologies. It is a global collaboration, hosted by The Linux Foundation, that includes leaders in finance, banking, Internet of Things (IoT), supply chains, manufacturing. and technology.

According to the IBM blog [21], Hyperledger can support approximately 3,500 transactions per second with a latency of less than one second. Thus, Hyperledger can be expected to support the requirement to be a peer-to-peer spam detection platform. However, Hyperledger is a permissioned blockchain and the relevance of the proof of authority consensus needs to be further evaluated for this use case.

C. Costs to incentivize miners:

Gas costs are constant, while their variance is also predictable based on the size of the variables, so the cost of the Ether required for maintaining the framework can be estimated easily. The cost of maintaining the system is discussed in section IV. There are two main topics that needs to be addressed here. First is source of money required to incentivize the miners and other is reducing the reporting frequency that reduces that overall costs. According to [32], spam callers earn a whooping amount of 9.5 billion USD per year by exploiting people using robocalls. Also, on an average, people spend 558 minutes per year attending these robocalls. That’s more than a full day of paid vacation. This loss incurred by the people due to spam calls can be invested as initial spending to this application. The spammers can be fined once people start reporting spam and the detection picks up and the amount can be diverted back as funds to incentivize the miners.

Improvements to consensus mechanism in Ethereum [20] will reduce the costs incurred by transactions as it replaces the proof of work consensus with proof of stake. Also reducing the spam reporting frequency by reporting spam calls once a day as a bulk update can help reduce overall costs.

D. Ledger access Times:

The read times of the phone number or domain ratings were achieved in real-time, which is a must for the blockchain to facilitate users to identify the phone number as a spammer when a call is received.

VI. LIMITATIONS AND FUTURE WORK

This approach of using a distributed ledger for peer to peer spam sharing has two main technical limitations, the concurrency of transactions and the storage size on the ledger. The transactions per second supported needs to be very high and the ledger size is going to increase in magnitude as more and more spam transactions are reported to the blockchain.

The methods discussed in [20][23][27][29][30] aim at improving the performance of blockchain using various techniques like sharding, improvements in consensus mechanisms etc., however, they need to be optimized to handle storage workloads of this magnitude. The sharding techniques need to be optimized in blockchains to have shards that are spawn over different locations so that the transactions in a region are processed locally reducing the latencies that are incurred due to communication between the shards. Multiple side chains per region connected via a mainchain that has metadata of all the side chains will be a good architecture for this kind of applications which have huge data storage requirements. Some of the methods mentioned in [31] like having shards that could have properties such as ultra-fast block times and cheaper transactions fees could be used for data publishing applications. Pruning the old data after a certain period can also help reduce the size of ledger effectively.

The storage problem can be handled by using Inter planetary file systems [11] to store the data while blockchain stores only the hash of the data. The blockchain still maintains the data about the reputation score while the original SPAM transactions can be offloaded to IPFS like file systems

Future work includes implementing optimized reputation-tracking algorithms in solidity code. R. Zhang et al. [12] discuss a novel approach for collaborative reputation-based voice spam filtering framework. Farideh Barghi et al. [28] propose an anti-SPIT (Spam over Internet Telephony) mechanism that calculates caller reputation based on multiple factors like call rate, call duration, and call patterns.

Additionally, reducing the spam reporting frequency to a value that can effectively reduce the number of transactions needs to be explored. And finally, other consensus mechanisms like Proof of Authority and Proof of Stake can be validated for the relevance of this use case.

VII. Acknowledgement

This material is based upon work supported by the National Science Foundation under awards 1241768 and 1637291.

References

[1] "Voice over IP." *AccessScience*, doi:10.1036/1097-8542.802030.

[2] "30 Billion Robocalls in 2017." *The YouMail Blog*, 17 Jan. 2018, blog.youmail.com/2018/01/30-billion/

[3] "Robocalls." *Consumer Information*, 13 Feb. 2019, www.consumer.ftc.gov/features/feature-0025-robocalls

[4] Tu, Huahong, et al. "SoK: Everyone Hates Robocalls: A Survey of Techniques Against Telephone Spam." 2016 IEEE Symposium on Security and Privacy (SP), 2016, doi:10.1109/sp.2016.27

[5] "Security Apps to Protect You & Your Phone - AT&T." *My ATT*, www.att.com/features/security-apps.html.

[6] "Call Protection | Automatic Scam Protection for Your Phone, T-Mobile, www.t-mobile.com/resources/call-protection.

[7] "Truecaller Is Transforming Today's Phonebook to Make It More Intelligent and Useful.." Truecaller, www.truecaller.com/.

[8] "Stop Unwanted Robocalls & Telemarketers with RoboKiller." RoboKiller, www.robokiller.com/.

[9] "Use Caller ID & Spam Protection - Phone App Help." Google, Google, support.google.com/phoneapp/answer/3459196?hl=en

[10] *National Do Not Call Registry*, www.donotcall.gov/.

[11] Juan Benet "IPFS - Content Addressed, Versioned, P2P File System https://github.com/ipfs/papers/raw/master/ipfs-cap2pfs/ipfs-p2p-file-system.pdf

[12] Zhang, Ruishan, and Andrei Gurtov. "Collaborative Reputation-Based Voice Spam Filtering." 2009 20th International Workshop on Database and Expert Systems Application, 2009, doi:10.1109/dexa.2009.95.

[13] Richards, Robert. "Representational State Transfer (REST)." *Pro PHP XML and Web Services*, 2006, pp. 633–672., doi:10.1007/978-1-4302-0139-7_17.

[14] HTC, www.htc.com/us/.

[15] "EXODUS." Genesis Block. EXODUS Phone, www.htcexodus.com/us/.

[16] "A One-Time Password System." IETF Tools, tools.ietf.org/html/rfc2289.

[17] Etherscan.io. "Chart Ethereum Pending Transactions Queue." *Ethereum BlockChain Explorer and Search*, etherscan.io/chart/pendingtx..

[18] Etherscan.io. "Chart Ethereum GasLimit History." *Ethereum BlockChain Explorer and Search*, etherscan.io/chart/gaslimit.

[19] "Ethereum Network Status." Ethereum Network Status, ethstats.net/.

[20] Ethereum."Ethereum/Wiki."GitHub, github.com/ethereum/wiki/wiki/Sharding-roadmap.

[21] "IBM Research: Behind the Architecture of Hyperledger Fabric." *The Analytics Maturity Model (IT Best Kept Secret Is Optimization)*, IBM Corporation, 7 Feb. 2019, www.ibm.com/blogs/research/2018/02/architecture-hyperledger-fabric/.

[22] "About – Hyperledger." Hyperledger, www.hyperledger.org/about.

[23] Mechkaroska, Daniela, et al. "Analysis of the Possibilities for Improvement of BlockChain Technology." 2018 26th Telecommunications Forum (TELFOR), 2018, doi:10.1109/telfor.2018.8612034.

[24] "ETH Gas Station." *ETH Gas Station*, www.ethgasstation.info/.

[25] Etherscan.io. "TESTNET Ropsten (ETH) Blockchain Explorer." *TESTNET Ropsten (ETH) Blockchain Explorer*, ropsten.etherscan.io/.

[26] Etherscan.io. "Chart Ethereum GasPrice History." Ethereum BlockChain Explorer and Search, etherscan.io/chart/gasprice.

[27] Decker, Christian, and Roger Wattenhofer. "Information Propagation in the Bitcoin Network." *IEEE P2P 2013 Proceedings*, 2013, doi:10.1109/p2p.2013.6688704.

[28] Barghi, Farideh, et al. "A Comprehensive SPIT Detection and Prevention Framework Based on Reputation Model on Call Communication Patterns." *2014 Iranian Conference on Intelligent Systems (ICIS)*, 2014, doi:10.1109/iraniancis.2014.6802606.

[29] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena. "*A Secure Sharding Protocol For Open Blockchains*". *In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS* '16, pages 17–30, New York, NY, USA, 2016. ACM

[30] Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ewa Syta and Bryan Ford. "*OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding* ". In the proceedings of *2018 IEEE Symposium on Security and Privacy, 2018 .doi*: 10.1109/SP.2018.000-5

[31] "Sharding FAQs · ethereum/wiki Wiki" https://github.com/ethereum/wiki/wiki/Sharding-FAQs

[32] "Robo Caller task force" https://www.robokiller.com/robocalltaskforce