Consensus Protocols for Blockchain-based Data Provenance: Challenges and Opportunities

Deepak K. Tosh[1], Sachin Shetty[2], Xueping Liang[3], Charles Kamhoua[4], Laurent Njilla[5]
[1]Department of Computer Science, Norfolk State University, Norfolk, VA
[2]Virginia Modeling Analysis and Simulation Center, Old Dominion University, Norfolk, VA
[3]College of Engineering, Tennessee State University, Nashville, TN
[4] Army Research Lab, Adelphi, MD
[5]Cyber Assurance Branch, Air Force Research Laboratory, Rome, NY
dktosh@nsu.edu, sshetty@odu.edu, xliang@tnstate.edu, charles.a.kamhoua.civ@mail.mil, laurent.njilla@us.af.mil

***Abstract*—Blockchain has recently attracted tremendous interest due to its ability to enhance security and privacy through a immutable shared distributed ledger. Blockchain's ability to detect integrity violations are particularly key in providing assured data provenance in cloud platform. The practical adoption of blockchain will largely hinge on consensus protocols meeting performance and security guarantees. In this paper, we present the design issues for consensus protocols for blockchain based cloud provenance. We present the blockchain based data provenance framework for cloud. We find that there are performance and security challenges in adopting proof-of-work consensus protocol within this framework. We present unique design challenges and opportunities in developing proof-of-stake for data provenance in cloud platform.**

***Index Terms*—Blockchain, cloud computing; Data provenance; Distributed consensus; Proof of work (PoW); Proof of stake (PoS)**

I. Introduction

Blockchain technology has attracted tremendous interest from a wide range of stakeholders, including finance, healthcare, utilities, real estate, and government agencies. Blockchains are shared, distributed and fault-tolerant database that every participant in the network can share, but no entity can control. Blockchain's distributed database maintains a continuously growing list of records, called blocks, secured from tampering and revision by distributed storage and continuous verification. The blocks contain a temporal listing of transactions that are stored in a public ledger using a persistent, immutable and append-only data structure that is globally accessible by every participant in the underlying peer-to-peer network.The technology is designed to operate in a highly contested environment where, adversarial strategies are nullified by harnessing the computational capabilities of the honest nodes such that information exchanged is resilient to manipulation and destruction. Tampering of blockchains is extremely challenging due to use of a cryptographic data structure and no reliance on secrets.

Blockchain utilizes a distributed consensus algorithm over a decentralized peer-to-peer network for verification of transactions prior to adding blocks to the public ledger. The verification process is determined by users and does not require a centralized administrator. The distributed consensus protocol ensures that the newly added transactions are not at odds with the confirmed transactions in the blockchain and maintains the correct chronological order. The newly added transactions which are waiting to be confirmed are packed in a block and submitted to the blockchain network for validation. In order for the block to be validated, the nodes in the peer-to-peer network solve a crypto-puzzle using computational resources at their disposal. The block is appended to the blockchain once a solution is discovered. This approach is known to be proof of work (PoW) [1] and it turns out to be an energy inefficient consensus approach [2] because the annual estimated electricity consumption of Bitcoin is 15.77 Terawatt hour, which is 0.08% of world's electricity consumption.

With PoW approach, miners opt for various specialized hardware to achieve their computational ability while at the same time they invest in electricity to operate and cool down these hardware. Knowing the eventual goal of the miners is to win the block-adding race so that they can be rewarded, a significant amount of energy is required to do so. The power that is spent to reach consensus using the PoW approach is mostly used in computing the irreversible SHA256 hashing function. Since the value of direct incentives will diminish eventually, the critical question of "how the PoW miners will be motivated to mine?" has to be addressed so as to smoothly run the consensus process. In addition to the energy wastage issue, there exists other security concerns which we describe in the next section. Since blockchain is turning out to be a robust tool to maintain an incorruptible distributed ledger, it has immense usefulness in cloud computing domain, especially in maintaining provenance of data objects across the cloud infrastructure. The existing PoW consensus can have additional overhead to maintain the blockchain in cloud domain. Therefore, it is of utmost importance to design and develop alternate form of consensus particularly for cloud provenance. In this paper, we study the concerns of PoW consensus and the advanced consensus protocols proposed to alleviate issues raised by PoW. We also expressed the need of provenance framework in cloud computing domain and provide an architecture for blockchain enabled provenance system, namely BlockCloud. Finally, we discuss the design

Approved for public release, distribution unlimited 88ABW-2017-4823, dated 28 September 2017.

challenges and opportunities in developing a proof of stake based cloud data provenance framework.

The paper is organized as follows. In Section II we present the PoW challenges and discuss usability of proof of stake consensus model. The Section III expresses the importance of data provenance in cloud systems. The architecture of PoS enabled blockchain cloud is presented in Section IV. Future research directions are briefed in Section V and Section VI concludes the paper.

II. PoW Challenges and Other Consensus Models

A. PoW Attack Concerns

Furthermore, the PoW mechanism's computational procedure for creating the ledger can be exploited by adversaries to impact the integrity of the blockchain. Recently, researchers have listed attacks which exploit the PoW consensus procedure, such as: selfish mining, 51% majority [3] manipulation attack, consensus delay [4] due to distributed denial of service, pollution log, blockchain forking, orphaned blocks, deanonymization, and block ingestion [5]. The impact of the attacks on the PoW mechanism are not felt the same for all blockchain applications. For instance, the 51 % majority manipulation attack is unlikely in cryptocurrency, but more likely in cloud platform, wherein adversaries can create collusion among several virtual machines across federated cloud providers to cause consensus delay.

The following are some blockchain anomalies that can stem for either attacks or random faults or both:

- Selfish mining - The selfish mining attacks are motivated by increasing returns for adversaries and impacting the fairness. The modus operandi for the attack involves adversaries selectively choosing the timing to publish discovered blocks. The intent of the attack is to maximize the chances of generating a longer blockchain than the rest of the network by consistently claiming block rewards for the batch of released blocks. These selfish mining attacks can invalidate blocks of honest nodes and negatively impact the reliability, fairness and robustness of the network.
- 51% majority manipulation - This attack is a coordinated effort by a group of adversaries to manipulate the blockchain network by controlling over a majority of the network's computational power. The major impacts of this attack are: 1) invalidate transactions/blocks at will by denying acceptance in the network; 2) equivocation of transactions and/or blocks by reversing the confirmation; 3) prevent other nodes from adding any blocks for different periods of time.
- Consensus delay - In this attack, adversaries inject false blocks and/or launch distributed denial of service to cause delays in reaching consensus in the blockchain. The impact of this attack on time-critical applications is devastating when consensus needs to be achieved within a short period of time.
- Blockchain fork - Blockchain forks are caused when nodes in the peer-to-peer network have diverging views about the state of the blockchain over long periods of time. These forks can be a result of accidental actions, such as, protocol malfunction and client software upgrades or intentional actions, such as, Trojan nodes that poison the validation process. Adversaries can exploit the presence of blockchain forks to create instability and mistrust among the nodes in the network.
- De-anonymization - The public availability of the blockchain transactions makes it possible to use data analytics techniques to analyze vast amounts of data within them. The analysis could provide valuable information that can sometimes reveal the individual transactions of participants and disclose their identity.

B. Proof of Stake (PoS) Consensus

To circumvent the problem, various consensus mechanisms, such as proof of stake [6], proof of activity [7], variants of Byzantine fault tolerant (BFT) algorithms [8], proof of space [9], are proposed that aim to avoid depletion of computational resources. Among those, the Proof of Stake (PoS) consensus protocol is an interestingly attractive one, which provides the block-inclusion decision making power to those entities that have stakes in the system irrespective of blockchain's length or history of the public ledger. The principal motivation behind this scheme is to place the power of leader-election in blockchain update process into the hands of the stakeholders. This is done to ensure that the security of the system will be maintained while the members stakes are at risk. Roughly speaking, this approach is similar to the PoW consensus except the computational part. Hence, a stakeholder's chances to extend the blockchain by including its own block depends proportionately on the amount of stake it has in the system. We can observe that the PoS consensus mechanism requires the stakes to be pre-distributed at the beginning of the process which was not the case with the PoW approach. However, we see an opportunity of exploiting the PoS based consensus to maintain data provenance blockchain in the cloud computing architecture since most cloud users have pre-acquired their necessary virtual computing resources for their operations.

The initial proof of stake (PoS) design included the age of cryptocurrencies and the total amount to define stake of each miner in the system. To gain the privilege of generating a PoS based block, the miner has to make a special coinstake transaction to himself so as to reset the coin age and prove that its stake is valid. According to their approach, a miner has a chance to extend the blockchain with his block having total unspent output $\mathcal{U}$, given the following condition is satisfied. Here, the unspent output refers to the output of a transaction that is not yet an input of another transaction, which means that the output is not been spent.

$$hash(hash(\mathbb{B}_{prev}), \mathcal{U}, t) \leq d \times balance(\mathcal{U}) \times age(\mathcal{U}) \quad (1)$$

where, $\mathbb{B}_{prev}$ is the previous block on which blockchain is to be extended, $balance(\mathcal{U})$ is the miner's stake amount, $age(\mathcal{U})$ is the aggregated age of the stake, and d is the mining

difficulty, which is of higher value unlike the traditional PoW based consensus. As seen in the Eq. 1, the computed hash value in the left side of inequality depends on the miners stake amount, so a large stakeholder can easily find a hash and hence has higher probability of adding its block in the blockchain. However, it is challenging to adopt this form of PoS in blockchain based cloud data provenance framework because the resources in cloud do not exhibit highly correlated characteristics with the tokens of cryptocurrency domain.

III. Data Provenance in the Cloud

Cloud computing environments are dynamic and heterogeneous and involve several diverse and disparate software and hardware components which are manufactured by different vendors and require interoperation. assurance of the ancestry of the data (where the data came from) is a challenge in cloud environments. Data provenance addresses the ancestry of the data based on detailed derivation of the data object. If true data provenance existed in the cloud for all data stored on cloud storage, distributed data computations, and data exchanges and transactions, then detecting insider attacks, reproducing research results and identifying the exact source of system or network intrusions would be achievable. Unfortunately, the state-of-the art in data provenance in cloud does not provide such assurances and there is a need to develop techniques to address this challenge.

Current state-of-the art provenance systems in the cloud support the above tasks through logging and auditing technologies. To identify the origin, cause and impact of security violations in cloud infrastructures will require collection of forensics and logs from the diverse and disparate sources which is an unduly heavyweight task. At the same time, logs only provide a sequential history of actions related to every application. The provenance data provides the history of the origins of all changes to a data object, list of components that have either forwarded or processed the object and users who have viewed and/or modified the object and has enhanced requirements for assurance. Besides the limited functionality of comparing logs to audit data, today's provenance functions are done in a private manner to establish ownership of digital assets. This, in turn, has a few limitations. First, the cost of provenance is high and prohibitive, in the sense that a provenance assurance should be established for each individual cloud service. Second, the process of provenance assurance, when multiple players are involved as is typical in cloud computing, lacks transparency. As such, moving to a more transparent, open, and public system is desirable.

A. Need of Blockchain in Maintaining Data Provenance

Blockchain technology provides such a capability and resolves many needed functionalities and properties for effective provenance in cloud [10]. In essence, a blockchain is a peer-to-peer ledger system, where information that constitutes provenance for physical, virtual, and application resources can be stored publicly for transparent verifiability and audit. As such, both transparency and cost effectiveness are provided, while access control and privacy for individual users of the ledger are ensured through encryption techniques, where individuals can see only parts of the ledger that is related to them.

Thus, blending the blockchain technology into the cloud environment can lead to achieve the task of data provenance, where the cloud nodes implicitly create a distributed network to record provenance data in the distributed and fault-tolerant ledger that is secured with a strong cryptographic notion. This distributed ledger of the blockchain is to be updated by all the nodes in cloud environment, but this depends on a certain rule that every node agrees upon. Designing such a consensus mechanism that ensures consistency in the blockchain is challenging. The traditional PoW consensus approach may not be applicable especially in the cloud computing domain due to its large computational power requirement. Therefore, it is important to investigate the usefulness of the proof of stake (PoS) based consensus method in this situation.

IV. Architecture and PoS Model for Blockchain Cloud (BlockCloud)

BlockCloud is our proposed data provenance architecture built on top of blockchain technology, which will provide the ability to audit data related operations for cloud users and providers in the federated cloud setup. It aims to achieve the following four objectives.

- Cloud Data Provenance: User operations are monitored in real time to collect provenance data to support access control policy enforcement [11] and intrusion detection.
- Proof-of-Stake Validation - As opposed to our previous PoW based provenance model [12], the consensus process BlockCloud will be driven by the staked resources of virtual machines housed in a federated cloud computing environment. The presence of validating VMs will provide supervisory control over the consensus process.
- Tamper-proof Environment: Data provenance record is collected and then published to the blockchain network which protects the provenance data. All data on the blockchain is shared among blockchain network nodes. BlockCloud builds a public time-stamped log of all user operations on cloud data without the need for a trusted third party. Every provenance entry is assigned a blockchain receipt for future validation.
- Provenance Data Validation: Data provenance record is published globally on blockchain network, where a number of blockchain nodes provide confirmation for every block. ProvChain uses blockchain receipt to validate every provenance data entry.

Our proposed BlockCloud architecture, as depicted in Figure 1, achieves the above objectives by monitoring user activities in real time using hooks and listeners (special classes of event listeners) so that every user operation on files will be collected and recorded for generating provenance data. Each piece of provenance information is referred as transactions that is broadcasted to the core of blockchain network created by a specific set of validating VMs. The validators collect the raw transactions and create their blocks individually, then

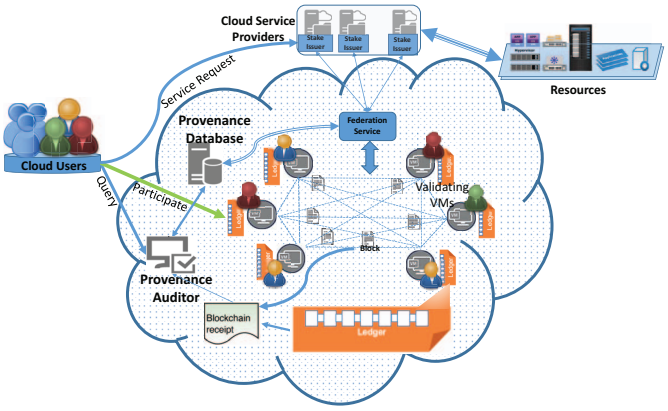

Fig. 1: BlockCloud Data Provenance Architecture

wait for the consensus to converge after which a leader will be selected to extend the blockchain. Every validator node on the blockchain network can verify the data operations or transactions before including into their block to ensure that the provenance data is authentic. We consider the presence of a provenance auditor in our model who collects the confirmation of each transaction block that is successfully added into the blockchain and records in the searchable provenance database. To maintain the privacy of cloud users, double hash of their user ID is used while broadcasting their transactions so that validators and provenance auditor cannot determine the exact user identities involved in those data operations.

In regard to the consensus process, we have considered the proof of stake model, where a selected set of validators among all the cloud users participate. To resolve the issues of initial resource distribution as well as stake validation, we assume a federation service in our model. This service starts at the beginning of consensus process and continuously work alongside the cloud validators to verify their stakes and select leader of every round in the blockchain network. The federation service is also considered to have the authority to decide the reward for successful block extension and at the same time to punish if validators act maliciously. Given the stakes of validators are used to decide leaders in the consensus process, they are not allowed to use the staked resources during the slot and it is monitored by the federation service. If it is found to be acting dishonestly, the staked resources will be forfeited. This will ensure that the cloud validators cannot abuse the proposed provenance system.

Now, we describe the role of BlockCloud's critical components as illustrated in the figure.

- Cloud User: A user in the cloud setup may have ownership over its own data and/or shared relationship on other users' data. They may participate in the PoS based consensus of provenance blockchain to help each other in creating a tamper resistant cloud environment. Participation in provenance may be voluntary or reward-driven but it comes at a cost of devoting some of their resources for achieving consensus.
- Cloud Service Provider (CSP): The cloud service provider offers a cloud storage/compute service and is responsible for user registration. Multiple CSPs form a federated environment that allows the cloud users to dynamically stake the virtual resources allocated to them. It is possible that they can be a part of the provenance system by participating in the blockchain consensus, which will benefit them in avoiding unauthorized data manipulations across the network.
- Provenance Database: The provenance database records all provenance information regarding every data object belonging to one or multiple cloud users. The provenance information is validated by the miners of the blockchain network, and is used for detecting malicious behaviors in the system. All data records are implicitly anonymized so as to protect cloud users' identity and privacy.
- Provenance Auditor (PA): PA can retrieve all the provenance data from the blockchain into the provenance database and validate the blockchain receipts. The PA maintains the provenance database but cannot link the provenance entry to the corresponding data owner. PA also acts as a mediator to query for provenance records from users so that every user can keep only the block headers with them and request the PA to fetch the detailed records whenever necessary.
- Blockchain Network: The blockchain network consists of a set of participating cloud users. This network only serves to run the consensus procedure by communicating transaction blocks with each other. This may not be a fully connected network as exhibited in the diagram, but can have a multi-hop topology at its core.
- Federation Service: This is a resource controlling entity that manages the process of stake allocation and verification. It also serves for deciding leaders in each block addition round based on the amount of staked resources.

Following the above architecture design, the blockchain enabled data provenance service with the proposed Proof of Stake based consensus in the cloud semantics is illustrated in Algorithm 1.

A. Stake in Cloud Computing Environment

From the cryptocurrency sense, the definition of stake is quite understandable, however, there is no prescribed way to correlate the definition of stake in a cloud system. So, we propose a stake model for the cloud system users as they participate in the consensus process. We consider the important resource components, such as CPU power, allocated memory, and network capabilities, as stakes for the cloud users, which are provisioned by the service provider in an on-demand basis.

B. Modeling Stake for a Cloud Instance

Considering the fact that users in a cloud environment are the entities, who occupy several virtual resources for their services and operations, such allocation of resources is tied to the idea of modeling stake for cloud users. For provisioning distributed consensus in the cloud environment, the stake must be defined in terms of the important resources a cloud user

Algorithm 1: Proof of Stake for BlockCloud

```
Data: TXs, Mining Peers
Result: New block generation and validation
initialization;
while true do
    peerNum ← number of peer nodes in the cloud;
    i ← 0;
    while i < peerNum do
        peer(i).stake = computeCloudStake();
        peer(i).possibility =
        computePossibility(peer(i).stake);
    end
    StakeWinner = StakeBiasedRandomChoose(peers);
    newBlock = StakeWinner.generateNewBlock(txs,
    StakeWinner.stake, prevHash);
    newBlock.broadcast();
    if peers.validate(newBlock) == true then
        ledger.append();
    else
        newBlock.discard();
    end
end
```

holds in the system. For simplicity, we consider the above three critical components/resources to define our stake model.

We consider participation in BlockCloud data provenance system to be either voluntary in nature or spurred by the incentive stemming from the possibility of getting rewarded in the future for maintaining the blockchain. To model the stake component, we assume a cloud user i has been allocated with total C_i number of CPU slices, S_i amount of storage in kilobytes, along with a data rate of D_i Kbps to perform its business operations and serve for maintaining the blockchain based data provenance. The above parameters are not the exhaustive list of resources that are needed to model the stake but it is still an open problem to come up with different innovative stake designs using several other cloud resources. The stake $\mathcal{X}_i = f(C_i, S_i, D_i)$ for user i can be defined as a function of above parameters, where, $f(.)$ is a transformation function of different cloud resources to a common scale that represents the stake of a cloud user i that is needed for the other members to compare and verify staked resources. Since the parameters of the function are homogeneous in scale across all the virtual miners in cloud, the function must be increasing in nature with respect to increase in quantity of each resource (C_i or S_i or D_i). Thus, $\frac{\partial \mathcal{X}}{\partial C_i} > 0$, $\frac{\partial \mathcal{X}}{\partial S_i} > 0$, and $\frac{\partial \mathcal{X}}{\partial D_i} > 0$ for each cloud miner i. This property is a necessary condition to satisfy because the cloud users must have a common ground to compare the stakes for understanding who has more stake in the system, so that leader election in consensus process will be easier to achieve. For instance, when two stake amounts from user i and j are compared and found that $\mathcal{X}_i \geq \mathcal{X}_j$, then it is inherent that $(C_i, S_i, D_i) \succcurlyeq (C_j, S_j, D_j)$ is satisfied, where $\succcurlyeq$ represents a component-wise comparison. In other words, user i has staked more resources compared to j.

V. Research Directions

Despite the advantages of PoS based blockchain in ensuring data provenance in intra-cloud and inter-cloud environment, there are several research avenues to be explored in implementing a robust proof of stake consensus model for the cloud computing systems.

A. Role of Cloud Service Provider (CSP)

Since cloud computing platforms are designed to provide quick, on-demand access to private/public resources and aim to meet the service level agreements (SLA) in a consistent manner, the role of a cloud provider is critical in such an environment so as to manage the dynamism of workload and resource utilization. Although an CSP is responsible for providing a secure, reliable and highly available cloud environment to the end users, it is unethical to track the users' internal service data that are used for ensuring provenance. Therefore, the blockchain-enabled data provenance system for the cloud must resolve the challenge of whether to keep the CSP as a part of the blockchain consensus process or leave it out. Since, we consider the provenance service to be an integral component toward security-provisioning of the cloud environment, inclusion of the CSP in the consensus process will maintain the blockchain decentralization intact. Moreover, the nature of the blockchain, whether permissioned or permissionless, to adopt in the cloud provenance still needs to be analyzed. Exclusion of the CSP from the blockchain consensus could make the cloud users insecure because the provider still possesses the control of data flow in the cloud environment. Thus, this trade-off is a crucial. It needs to be resolved while designing the PoS based blockchain for cloud data provenance.

B. Initial Resource Distribution

To begin the blockchain consensus process in a cloud environment, the users first need to have an understanding of the amount of resources each user holds in order to verify their stakes in the system at later stages. In case of crypto-currency, this process involves members purchasing the bitcoins from an external agency and transferring to their wallet before they perform any transactions. However, when a similar mechanism is adopted in the design of BlockCloud, it poses the concern of whether to rely on the external provider or not because it may not be a part of the provenance system. Apart from this, revealing the distribution of resources among the peers (irrespective of inter/intra-cloud users) may disclose their stake-power, which can be of particular interest when a malicious cloud miner is co-resident. Thus, a robust mechanism is required to distribute the initial resources so that cloud users' privacy and secrecy, related to resource utilization, remains unaltered.

C. Amount to Stake

Considering the fact that cloud users have been provisioned resources to use the services, a portion of the provisioned resources must be reserved to enable PoS based consensus for the BlockCloud. For the consensus process operating in time-slotted implementation, it is assumed that the staked resources are hindered from running any user-level services for a particular slot unless consensus is achieved. However, at the end of each slot, cloud users have the option to modify the staked amount prior to proceeding to the next consensus slot. Holding up the staked resources will be helpful for avoiding untruthful behavior of users in the system, so that when any maliciousness is detected, the corresponding user will forfeit its stake. Although higher stakes in the system provides user a high chance to add its block and collect incentives from the transaction fees. There is still an open question: *What amount of resources can a user choose to dedicate for the consensus process so that it balances out the gains from service provisioning of such resources and participation in BlockCloud consensus?*

D. Incentivizing Cloud Stakeholders

Another important challenge in the PoS enabled BlockCloud is the identification of incentive mechanism to motivate the cloud users to participate in the consensus process. Serving in the blockchain consensus requires the users to dedicate some of their resources to serve for the rest of the cloud users. Hence, sufficient reward for the participants needs to be provisioned, else they may leave the framework. The reward component in the PoW based blockchain was to incentivize the winner with a specific amount of coins. However, monetary reward may not be feasible in the cloud computing environment. Thus, it is important to identify ways of rewarding the cloud stakeholders who play a major role in the consensus process. A possible solution to incentivize the cloud users can be the following: A certain percentage of the previously staked resources will be returned back to the user which can be used to run their services. Hence, if a user has higher stakes and truthfully participates in the blockchain consensus, it can effectively augment its computing power for usage toward serving its regular workload. However, for this scenario, an important design issue will be to identify the appropriate percentage of the stake to reward for survival of the BlockCloud because virtual resources cannot be created out of thin air.

E. Transaction Privacy

The blockchain involved in traditional crypto-currency achieves consensus by a set of miners who always compete among each other to extend the chain with their own block. Equivalently, there exists a set of privileged cloud users who are involved in maintaining consensus in the BlockCloud architecture. However, these virtual miners live in a federated cloud environment to perform the validity of cloud data operations by reaching to a common knowledge from the prior transaction details derived out of blockchain. Unlike the traditional blockchain where transactions refer to transfer of coin ownership, the transactions in BlockCloud includes records of various operations on different data objects created/shared by a single/multiple cloud users. Due to this, verification of coin ownership and validity of transactions are easier to perform in the traditional blockchain system, but in the case of BlockCloud, the actual stake and transactions included in a block are different. Hence, it is challenging for the virtual miners in the cloud environment to maintain the privacy of transactions while achieving consensus. So, the PoS consensus in BlockCloud will be negatively impacted unless there is a mechanism in place to derive the transaction history of various cloud data objects in cloud without hampering the privacy of data transactions performed by cloud users.

VI. Conclusions

Preventing malicious activities in the federated cloud environment requires assurance of data provenance so that every operation on data objects can be tracked effectively. Blockchain offers unique set of features to do so, however the underlying PoW consensus is inapplicable while integrating in cloud domain. We discussed the reasons which make PoW unfit for our case. We have proposed blockchain based cloud data provenance framework, namely BlockCloud, that incorporates PoS as consensus engine. Despite the benefits of PoS powered blockchain, we came across several design challenges while integrating the distributed ledger technology in cloud computing domain, which we discussed in details.

References

[1] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system (2008).

[2] Bitcoin energy consumption index, http://digiconomist.net/bitcoin-energy-consumption.

[3] I. Eyal, E. G. Sirer, Majority is not enough: Bitcoin mining is vulnerable, in: International Conference on Financial Cryptography and Data Security, Springer, 2014, pp. 436–454.

[4] J. Göbel, H. P. Keeler, A. E. Krzesinski, P. G. Taylor, Bitcoin blockchain dynamics: The selfish-mine strategy in the presence of propagation delay, Performance Evaluation 104 (2016) 23–41.

[5] D. K. Tosh, S. Shetty, X. Liang, C. Kamhoua, K. Kwiat, L. Njilla, Security implications of blockchain cloud with analysis of block withholding attack, in: International Symposium on Cluster, Cloud and Grid Computing, IEEE/ACM, 2017.

[6] S. King, S. Nadal, Ppcoin: Peer-to-peer crypto-currency with proof-of-stake, self-published paper, 2012.

[7] I. Bentov, C. Lee, A. Mizrahi, M. Rosenfeld, Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract] y, ACM SIGMETRICS Performance Evaluation Review 42 (3) (2014) 34–37.

[8] M. Castro, B. Liskov, Practical byzantine fault tolerance and proactive recovery, ACM Transactions on Computer Systems (TOCS) 20 (4) (2002) 398–461.

[9] S. Dziembowski, S. Faust, V. Kolmogorov, K. Pietrzak, Proofs of space, in: Annual Cryptology Conference, Springer, 2015, pp. 585–605.

[10] S. Shetty, V. Red, D. Satterfield, C. Kamhoua, K. Kwiat, L. Njilla, Data provenance assurance in cloud using blockchain, in: SPIE Defense + Security Conference, 2017.

[11] J. P. D. Nguyen, R. Sandhu, Dependency path patterns as the foundation of access control in provenance-aware systems, 2012.

[12] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, L. Njilla, Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability, in: International Symposium on Cluster, Cloud and Grid Computing, IEEE/ACM, 2017.