Differentially Private Proof of Stake

Pranay Jain, Rathi Kashi, David Pujol

November 2021

Abstract

Proof-of-Stake (PoS) is a mechanism for block proposal through leader election based on the party's *stake* in the system. Current PoS-based protocols inadvertently disclose information about the amount of stake a validator has in the system. This is equivalent to discovering the wealth of the validators in the system, which is a breach of privacy and could potentially open up the network to selective attacks. We propose a *Differentially Private Proof of Stake* mechanism that anonymizes the stake for individual validators in the system. Our method performs leader election using a noisy stake in each round, which provides privacy that does not degrade over time. Through simulations, we show that our mechanism makes it hard to recover one's stake while ensuring that the fairness in leader election is not affected over a sufficiently long time.

1 Introduction

Modern day blockchain protocols rely on lottery based consensus mechanisms in order to elect leaders who are able to add blocks onto the blockchain. Proof of Work [15] is one such mechanism where a party's probability of being elected as leader is proportional to their share of total computing power in the system. By tying any individuals election probability to a scarce resource the system avoids the possibility of Sybil attacks, where a user inflates their probability of being elected. Proof of work achieves this goal by having parties solve computationally hard problems therefore wasting computing power and other associated resources.

As a response Proof of Stake [11, 6, 5] has arisen as a consensus protocol which uses significantly less resources. Instead of depending on computing power, it uses stake within the system as a scarce resource. Proof of stake mechanism assumes that the blockchain contains information about the stake a user has in the system. In the case of cryptocurrency systems this stake is often the amount of wealth the user has stored in that particular cryptocurrency. The user's then run a lottery with the the property that the probability of winning the lottery is proportional to the the stake that particular user has in the system.

The voting mechanism for Proof-of-Stake is also fundamentally different. In Proof-of-Work, a voting party can easily verify the validity of a block by computing the hash of the block along with the nonce included in it [15]. In Proof-of-Stake however, a block proposer must prove in some way that it is a leader at the time it is proposing the block. There are different implementations for such a verification mechanism, but they require the stake held by all parties to be public knowledge in order to verify that the block proposer was indeed elected. This is a privacy concern because the stake is equivalent to the wealth one holds in the system and many validators may not agree with sharing such information publicly. Furthermore, in Proof of Stake systems, the stake one has in the

system is directly related to how much security they provide to that system. Someone with a large amount of stake is at greater risk of attacks since it is easier for an adversary to affect security by attacking an honest party with a large stake than attacking the network as a whole. So, an adversary can mount a selective attack on such a party via malware or a denial-of-service, either to steal their secret key, or to prevent this party from functioning correctly [14].

It is, therefore, important to develop an Anonymous Proof of Stake protocol that elects leaders in proportion to their stake without disclosing their stake publicly. Moreover, such a system should also be resistant to a *frequency linkage attack*, where the adversary is able to reconstruct the stakes of all individuals in the system after enough runs of the mechanism by simply counting the frequency with which they propose blocks [7]. In this paper, we propose Differential Privacy [9], a method of protecting the privacy of individuals by adding noise to the proof of stake mechanism. The rest of the paper is structured as follows. In section 2, we will discuss current implementations of PoS election mechanism and related works on anonymous proof-of-stake protocols. In section 3, we formally introduce Differential Privacy and the theoretical guarantees that it provides. In section 4, we formally describe the implementation of Differential Privacy in the leader election mechanism. Section 5 will discuss results from simulation experiments based on our protocol design. Finally, we will conclude with a discussion on the benefits and potential limitations of our approach in Section 6.

2 Related Works

Proof-of-Stake was initially introduced in online forums and blog posts with the crypto community [1, 2]. The Proof-of-Stake lottery mechanism can be informally described as follows: there exists a publicly verifiable randomized mechanism that assigns each party in the system with an assignment which allows them (and anyone else) to determine whether they are eligible to propose a new block at a given time. Most implementations divide time into granular slots, called *rounds*, and leaders are determined independently for each round.

The process for leader election have two main implementations: *leader-based* and *committee-based* election. Algorand [11] elects a committee of some fixed size, and the committee runs a Byzantine Agreement subprotocol to agree on a block. On the other hand, Ouroboros [13] and SnowWhite [5] are protocols that elect one or more leaders for each round responsible for extending the blockchain. Modern implementations including Ouroboros Praos [6] and Algorand [11] use cryptographic Verifiable Random Functions (VRF) that allow parties to locally determine whether or not they win the lottery in a particular round. This is a salient feature that is required to implement privacy in a PoS protocol, although it falls short of the goal in these protocols since they require the stake distributions to be publicly known.

There have been some recent attempts at implementing Privacy-preserving Proof-of-Stake protocols. Ouroboros Crypsinous [12] is an improvement upon Ouroboros Genesis [3] inspired by ZeroCash [4], which is a privacy-preserving Proof-of-Work Protocol. However, Crypsinous does not address the frequency-linkage attack. Ganesh et al. [10] propose a Private PoS protocol that aims to solve the frequency linkage attack. For each party, it splits up the stake between several virtual parties; the party is deemed as a leader if any of its virtual parties' VRF results wins the lottery. However, both these protocols suffer from a potential anonymity attack that requires the adversary to control the network delay incurred by the parties up to a maximum threshold. This is in line with the synchronous network model that these protocols are designed for. The attack, described in [14], involves the adversary maintaining a no-delay channel between itself and its victim, while ensuring maximal delay between the victim and other honest parties. After that, if the adversary sends any

transaction to the victim, the transaction will be included in the next block only if the block was created by the victim. This way it can determine over time how often a block is proposed by its victim and deanonymize its stake.

To our knowledge, ours is the first work that introduces Differential Privacy to provide privacy guarantees in a Proof-of-Stake protocol. We note that in any privacy-preserving protocol, there is a potential broadcast channel attack where the adversary may discover information about one's stake by simply listening to the broadcast messages in the network. We do not provide an explicit solution to that problem and refer the reader to the discussion on Ideal anonymous broadcast channels in [14]. For the purposes of this paper, we assume to be operating in such an ideal broadcast channel.

3 Differential Privacy

Differential Privacy [9, 8] is a formal model of privacy that grantees each individual that any query computed from sensitive data would have been almost as likely as if the individual had opted out. More formally, Differential Privacy is a property of a randomized algorithm which bounds the ratio of output probabilities induced by changes in a single record.

Definition 1 (Differential Privacy)**.** *A randomized mechanism $\mathcal{M}$ is (ϵ)-differentially private if for two neighboring databases D, and D' which differ in at most one row, and any outputs $O \subseteq Range(\mathcal{M})$:*

$$\frac{\Pr[\mathcal{M}(D') \in O]}{\Pr[\mathcal{M}(D) \in O]} \leq e^{\epsilon}$$

The parameter ϵ often called the privacy budget quantifies the privacy loss. Here we focus exclusively on ϵ-Differential Privacy, i.e when $\delta = 0$.

The Laplace Mechanism is a differentially private primitive which underlines the algorithms used here. We describe the vector version of the Laplace Mechanism below.

Definition 2 (Laplace Mechanism)**.** *Given a query q and a datset x, the randomized algorithm which outputs the following query is ϵ-differentially private [9].*

$$q(x) + \text{Lap}\left(\frac{\Delta q}{\epsilon}\right)$$

Where Δq is the maximum change in q due to the change in one record (often called the sensitivity of q) and $\text{Lap}(\sigma)$ denotes a single sample from a Laplace distribution with mean 0 and scale σ.

Differentially private releases compose with each-other in that if there are two private releases of the same data with two different privacy budgets the amount of privacy lost is equivalent to the sum of their privacy budgets. More formally we have the following.

Theorem 1 (DP sequential composition [9])**.** *Let $\mathcal{M}_1$ be an ϵ_1-differentially private algorithm and $\mathcal{M}_2$ be an ϵ_2-differentially private algorithm. Then their combination defined to be $\mathcal{M}_{1,2}(x) = (\mathcal{M}_1(x), \mathcal{M}_2(x))$ is $\epsilon_1 + \epsilon_2$-differentially private*

Likewise multiple differentially private releases of different data compose with each other in that if there are two differentially private releases of different data with two different privacy budgets ϵ_1 and ϵ_2 the release of both sets of data satisfies differential privacy with privacy budget $\max(\epsilon_1, \epsilon_2)$. More formally as follows.

Theorem 2 (DP parallel composition [9])**.** *Let $\mathcal{M}_1$ be an ϵ_1-differentially private algorithm and $\mathcal{M}_2$ be an ϵ_2-differentially private algorithm each on a disjoint dataset. Then their combination defined to be $\mathcal{M}_{1,2}(x) = (\mathcal{M}_1(x), \mathcal{M}_2(x))$ is $\max(\epsilon_1, \epsilon_2)$-differentially private*

For our system we'll be leveraging the work of Narayan et. al. [16]. In this work the authors create a system which allows for answering queries under Differential Privacy in a verifiable way without revealing the underlying data. The authors create a special query language which will execute a query if and only if the query is differentially private (for some parameter ϵ) and will output that value along with a zero knowledge proof verifying that the output is differentially private and is using the data in question. This ensures that an adversary cannot simply re-sample noise until they get a noise value that is favorable to them. In the same vein this also ensures that an adversary cannot output a differentially private value to some fabricated data which is advantageous to them.

4 Differentially Private Proof of Stake

Here we present a method for using Differential Privacy to protect the stake values of any individual during the Proof of Stake Process. In this case we will consider the local model of Differential Privacy where each participant adds noise to their own data prior any computations done on it. We consider this mechanism to be an additional step that can be added to any existing proof of stake mechanism. Prior to electing a leader in the round each participant will run their stake through the private mechanism and submit the noisy stake to the POS mechanism. In order to do this in a decentralized manner we can use the work of Nayan et. al. [16] so that participants may report their noisy stake with an additional zero knowledge proof that the correct noise distribution was used as well as the correct initial stake, all while never revealing the original stake in question. In order to do so each participant will locally store their noisy transaction history as well as current transactions. This will allow the participant to write their stake in the next round as a simple sum query in the Verifiable DP query language. From there the Verifiable DP compiler will run both the mechanism as well as construct the zero knowledge proof. This allows the differentially private mechanism to be added to any existing proof of work protocols as an additional process before the election.

We currently consider protecting privacy at the transaction level, that is the output of our mechanism should be similar with the addition or removal of a single transaction. We consider this level of privacy protection to be a good balance between overall utility and privacy guarantee. One could also consider bounding the contribution of a single round of transactions but this would incur too much variance over a single round and could cause users that experience extremely high or extremely low stakes frequently.

4.1 Differentially private Mechanism

Before discussing the implementation of Differential Privacy we first must go over the additional restrictions to put onto a proof of stake system. Previous proof of stake systems [5] suggest putting a cap on the amount of stake that can be transferred in between individual rounds. In addition in order to bound the sensitivity of proof of stake we will impose an additional upper bound on the amount of stake that can be transferred per transaction. We'll call that bound α.

In order to use the zero knowledge proofs for Differential Privacy [16] we model the stake at each round as a series of sum queries. The first round of stake is simply a noisy reporting of the stakes at the genesis block (or whenever in the chain DP is implemented). Every round there after the stake reported is the stake reported at the previous round plus the sum of all transactions in the current round. In order to make this process differentially private we add Laplace noise, with scale $\frac{2\alpha}{\epsilon}$ to both the initial stake and the sum of transactions at each round.

Additionally we employ one additional step that reduces the variance in stake reporting over the long term. We split the rounds of POS into epochs each epoch consisting of a pre-specified number

of rounds. During each epoch we report stake using the previously stated mechanism, however at the end of an epoch the sum of noisy transactions is replaced with a single noisy sum query, with scale $\frac{2\alpha}{\epsilon}$, over all the transactions in the epoch. For epoch lengths which are significantly large this can decrease the variance dramatically as the variance introduced by the sum of many random variables are instead replaced with the variance of a single random variable of the same scale.

4.2 Proof of Differential Privacy

At each round including the genesis round we add Laplace noise, with scale $\frac{2\alpha}{\epsilon}$ to the reported stake. This ensures that each round satisfies $\frac{\epsilon}{2}$-Differential Privacy. Since the transactions at each round (as well as the genesis block) are independent of each-other the combination of all the private releases satisfies $\frac{\epsilon}{2}$-Differential Privacy by parallel composition. At the end of each epoch we also release the sum of all rounds in the epoch using Laplace noise with scale $\frac{2\alpha}{\epsilon}$. Since the end of epoch release and the releases during the epoch share data these compose using sequential composition and thus combined satisfy ϵ-Differential Privacy.

By using the variance of the Laplace distribution we can find that the expected L2 error at any given point in time is $(\lambda + \kappa + 1)(\frac{8\alpha^2}{\epsilon^2})$, where λ is the number of epochs that have passed and κ is the number of rounds that have passed in the current epoch.

4.3 Additional risks

We have identified two additional risks with naive implementation of DP proof of stake. The first risk we call a *Stakeless Elector*. A stakeless elector is a validator who has either no, or very little stake in the system but is elected as a leader for a particular round due to the inherent noise added by differential privacy. A fundamental law of differential privacy is that every outcome will happen with some non-zero probability [9] so this event is inevitable in any differentially private system. This introduces a new vector for Sybil attacks where an adversary creates many agents with little or no stake in order to increase their chances of being a stakeless elector. We propose imposing a minimum threshold of how much stake an agent must have before being allowed to participate in the election process. This is also the approach adopted by Ethereum 2.0 by imposing a minimum stake of 32 ETH to become a validator [17]. This will not eliminate the problem but will instead mitigate it. We will show through simulation the chances of having a stakeless elector and how the setting of a lower bound affects an adversaries ability to create a stakeless elector.

The second additional risk we have identified is that of the *Accidental majority*. This is the case when the noise introduced by differential privacy would cause one agent to become a majority stakeholder despite their actual share of the stake. Just like the case of the stakeless elector because of the inherent randomness of differential privacy this event must happen with some non-zero probability. We ran several simulations of our DP proof of stake mechanism in order to assess the probability of such events and possible mitigation mechanisms.

5 Experiments and Results

This section discusses the results of our simulation experiments. We simulated the Differentially Private Proof of Stake mechanism over many rounds and epochs. In our simulations, the change of stake is modeled as a round-robin pattern; that is, validator 1 sends X units to validator 2 in round r, validator 2 sends X units to validator 3 in round $r + 1$ and so on in a circular pattern. This was done so that over time none of the validators' accumulate stake by the transactions, while still

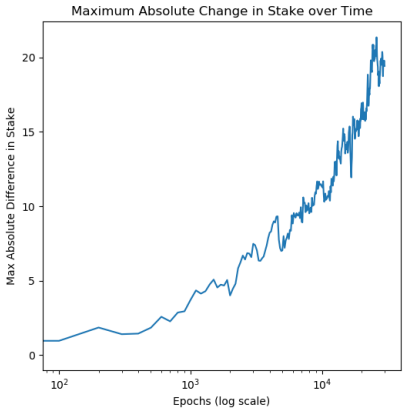

Figure 1: Stake growth over time

ensuring that some stake is changing hands in each round. We set $X = \alpha$, the maximum amount of stake allowed to change in the system. For these experiments, we set $\alpha = 0.1$ units of stake.

5.1 Stake growth over time

In order to evaluate the stake growth over time, as a result of the noise, we ran a simulation with 100 nodes starting with stake 32 each (the choice of starting stake was arbitrary). We simulated the accumulation of noisy stake values for 10^6 epochs with each epoch consisting of 1000 rounds. The graph obtained in Figure 1 indicates that the absolute change in stake from the original value over epochs. It's evident from the curve that for the first 1000 epochs, the maximum change is less than 15%, while at around $10,000$ epochs it reaches around 25%. The length of an epoch depends on the blockchain implementation. As an example, with Ethereum's current rate of 1 block per 12 seconds, 1000 epochs is around 5 months, while $10,000$ epochs is around 3.5 years.

5.2 Leader election frequency over time

In this experiment, we observe the frequency with which a validator is elected as leader. Ideally a validator should be elected with frequency *close to* their proportion of stake in order to maintain fairness, but enough for an adversary to recover the stake by a frequency linkage attack. In our experiment, we monitor a validator with 20% of the stake - the results are shown in Figure 2. Over several runs of the protocol, we see that the election rate for this node is consistently within 20% of its true share of stake (ie. $\pm 4\%$ change in rate of being elected) over a sufficiently long time frame. This indicates that the noise added does not adversely impact the fairness of protocol execution but masks the value of the true stake of the individual.

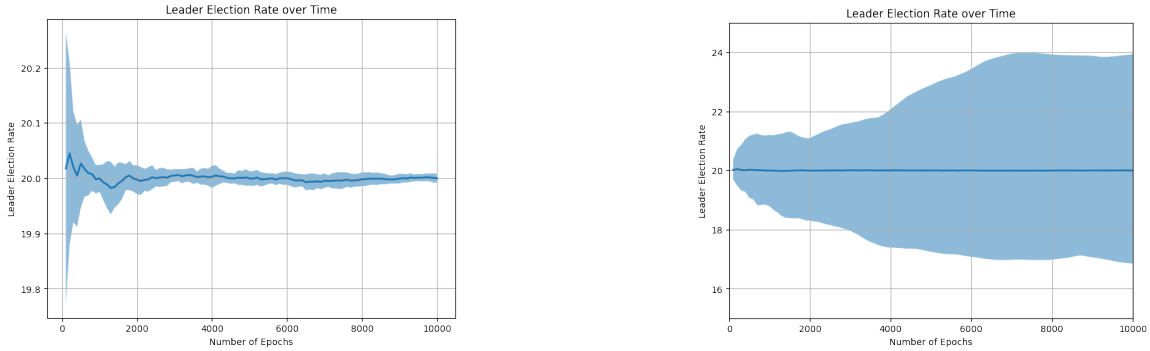

Figure 2: Leader election frequency over time (Epoch size: 1000 rounds, Number of Epochs: $10,000$). The figure on the left shows leader election frequency for a single in a typical Proof-of-Stake system. For a node with 20% of stake, it gets elected as leader very close to 20% of the time. On the right, we see the leader election frequency over time in the Differentially Private PoS system. There is higher variance in its election frequency which makes it difficult to uncover the party's actual stake.

The following experiment observes the variance in the frequency with which all the validators in the system get elected in a single run. We started 5 nodes with the following percentage of stakes: $[60, 20, 10, 5, 5]$ and tracked the percentage of times they were elected as leader at the end of each epoch. Figure 3 demonstrates the results. It's evident that leader election is still proportional to the stake of each party with some variance.

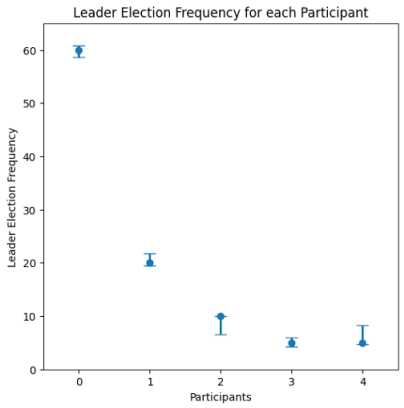

Figure 3: Leader election frequency over time with noise

5.3 Accidental Majority

The following simulation was run to identify whether accidental majority is a problem with the addition of noise: Starting with two different distribution of stake values, the probability of a non majority validator obtaining the majority was identified over time.

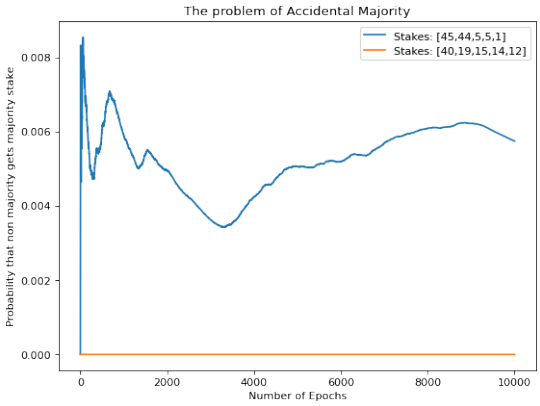

Figure 4: Probability of non majority validator gaining majority stake (Epoch size: 1000 rounds, Number of Epochs: $10,000$).

At the end of 10,000 epochs, the probability of an accidental majority came out to be 0.005 in the extreme case when the stakes were very close to each other. For stakes further apart, the probability came out to be 0.

5.4 Stakeless Elector

In order to verify the effect of a sybil attack on the network (which arises as a result of the stakeless elector problem), the following simulation was run: starting with an initial stake distribution, one of the validators splits their stake into a number of smaller stakes and the probability of the validator being elected as the leader is observed as the number of splits increase.

Figure 5 indicates that splitting the stake does translate to a higher probability of being elected as leader as a result of the noise added. Imposing a threshold on the amount of stake reduces the chances of the attack. A favorable threshold was found to be 6%. In practice, with multiple validators in a network, the threshold percentage will decrease to a more realistic amount.

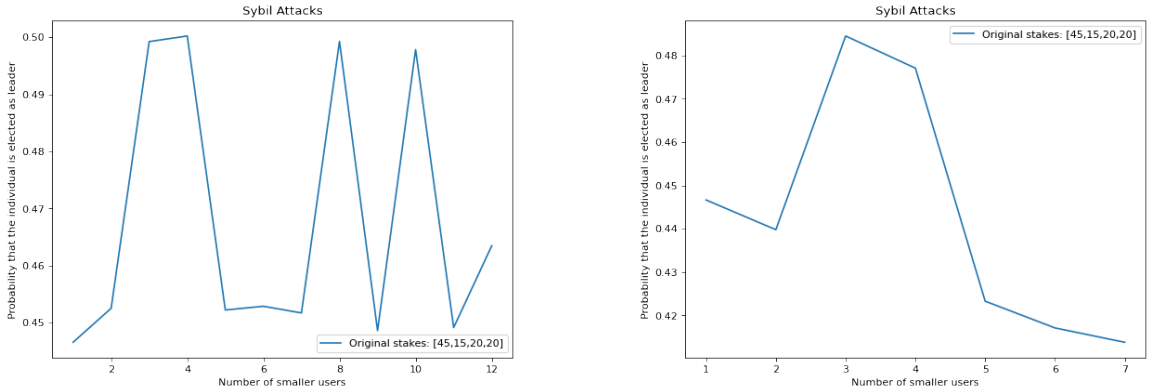

Figure 5: Probability of a validator being elected as the leader as the number of splits increase. The figure on the right is when a minimum threshold of 6% is added for a node to be able to join the network.

6 Discussion

In this paper we present a Differentially Private Proof-of-Stake mechanism. The goal of this mechanism is to provide reasonable guarantees of privacy to the stakeholders in the system. An ideal system should protect the knowledge of the stake in the system while ensuring fairness in the leader election process. We conducted simulation experiments to demonstrate empirically that such a system can be achieved using our Differentially Private mechanism.

The design of our mechanism ensures that it could be implemented in existing Proof of Stake blockchains. An implementation would require an initial preprocessing step, and parties will need to append an additional Zero Knowledge proof to the proof submitted as part of the leader verification process. Our current approach requires parties to publish a noisy version of their stake publicly. However, it may be possible to combine the Zero Knowledge Proof of DP with the Anonymous Verifiable Functions proposed in [10] such that a validator could provide a ZKP that proves their commitment to a noisy amount of stake. We leave this for future exploration. It is also possible to include this mechanism along with the approach proposed by Ganesh et al. [10] whereby we split a party's stake into virtual parties possessing smaller amounts of (noisy) stake. This will provide privacy guarantees to that system along with robustness against frequency linkage attacks.

We note that the accumulation of noise in some validators is a point of concern in this mechanism. Figure 1 demonstrates that any significant accumulation of noise would only happen after a very long time period (potentially on the order of years), it may still incentivize participants to exploit that. A potential attack is a kind of a grinding attack where an adversary keeps shifting their stake between new addresses until they get an initial series of lucky rounds where their stake is bumped up. It is possible to disincentivize such behaviour by introducing fees or penalties on unstaking or shifting stake too rapidly. We also show that is possible to control this effect by tuning various parameters of the protocol, such as the length of an epoch, transaction limit in a round, etc.

Differential Privacy is a state-of-the-art methodology for providing privacy guarantees on sensitive data. We believe Differentially Private Proof of Stake can provide the groundwork for implementing privacy in PoS blockchains and can be extended and combined with existing works to improve their privacy properties.

References

[1] I. Bentov et al. *Cryptocurrencies without proof of work*. In Financial Cryptography Bitcoin Workshop. 2016.

[2] I. Bentov et al. "Proof of activity: Extending bitcoin's proof of work via proof of stake". In: 2014.

[3] Christian Badertscher et al. "Ouroboros Genesis: Composable Proof-of-Stake Blockchains with Dynamic Availability". In: CCS '18. Toronto, Canada: Association for Computing Machinery, 2018, pp. 913–930. ISBN: 9781450356930. DOI: 10.1145/3243734.3243848. URL: https://doi.org/10.1145/3243734.3243848.

[4] Eli Ben Sasson et al. "Zerocash: Decentralized Anonymous Payments from Bitcoin". In: *2014 IEEE Symposium on Security and Privacy*. 2014, pp. 459–474. DOI: 10.1109/SP.2014.36.

[5] Phil Daian, Rafael Pass, and Elaine Shi. *Snow White: Robustly Reconfigurable Consensus and Applications to Provably Secure Proof of Stake*. Cryptology ePrint Archive, Report 2016/919. https://ia.cr/2016/919. 2016.

[6] Bernardo David et al. "Ouroboros Praos: An Adaptively-Secure, Semi-synchronous Proof-of-Stake Blockchain". In: *Advances in Cryptology – EUROCRYPT 2018*. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Cham: Springer International Publishing, 2018, pp. 66–98. ISBN: 978-3-319-78375-8.

[7] Irit Dinur and Kobbi Nissim. "Revealing information while preserving privacy". In: *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems - PODS 03* (2003). DOI: 10.1145/773153.773173.

[8] Cynthia Dwork. "Differential Privacy". In: *Automata, Languages and Programming*. Ed. by Michele Bugliesi et al. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006.

[9] Cynthia Dwork and Aaron Roth. "The Algorithmic Foundations of Differential Privacy". In: *Found. Trends Theor. Comput. Sci.* (2014).

[10] Chaya Ganesh, Claudio Orlandi, and Daniel Tschudi. *Proof-of-Stake Protocols for Privacy-Aware Blockchains*. Cryptology ePrint Archive, Report 2018/1105. https://ia.cr/2018/1105. 2018.

[11] Yossi Gilad et al. *Algorand: Scaling Byzantine Agreements for Cryptocurrencies*. Cryptology ePrint Archive, Report 2017/454. https://ia.cr/2017/454. 2017.

[12] Thomas Kerber et al. *Ouroboros Crypsinous: Privacy-Preserving Proof-of-Stake*. Cryptology ePrint Archive, Report 2018/1132. https://ia.cr/2018/1132. 2018.

[13] Aggelos Kiayias et al. "Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol". In: *Advances in Cryptology – CRYPTO 2017*. Ed. by Jonathan Katz and Hovav Shacham. Cham: Springer International Publishing, 2017, pp. 357–388.

[14] Markulf Kohlweiss et al. *On the Anonymity Guarantees of Anonymous Proof-of-Stake Protocols*. Cryptology ePrint Archive, Report 2021/409. https://eprint.iacr.org/2021/409.pdf. 2021.

[15] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. Accessed: 2015-07-01. Dec. 2008. URL: https://bitcoin.org/bitcoin.pdf.

[16] Arjun Narayan et al. "Verifiable differential privacy". In: *Proceedings of the Tenth European Conference on Computer Systems, EuroSys 2015, Bordeaux, France, April 21-24, 2015*. Ed. by Laurent Réveillère, Tim Harris, and Maurice Herlihy. ACM, 2015, 28:1–28:14. DOI: 10.1145/2741948.2741978. URL: https://doi.org/10.1145/2741948.2741978.

[17] *The Beacon Chain Ethereum 2.0 explainer you need to read first*. 2020. URL: https://ethos.dev/beacon-chain/.