IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 18, 2020, accepted September 24, 2020, date of publication September 30, 2020,
date of current version October 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3027952

A Novel Methodology-Based Joint Hypergeometric Distribution to Analyze the Security of Sharded Blockchains

ABDELATIF HAFID[1,2], (Member, IEEE), ABDELHAKIM SENHAJI HAFID[2], (Member, IEEE), AND MUSTAPHA SAMIH[1]

[1]Team of EDA – Mathematical Laboratory and their Applications, Department of Mathematics, Faculty of Sciences, Moulay Ismail University, Meknes 50050, Morocco
[2]Montreal Blockchain Laboratory, Department of Computer Science and Operational Research, University of Montreal, Montreal, QC H3C 3J7, Canada

Corresponding author: Abdelatif Hafid (a.hafid@edu.umi.ac.ma)

This work was supported in part by the Mohammed VI Polytechnic University - UM6P.

ABSTRACT Cryptocurrencies (e.g., Bitcoin and Ethereum), which promise to become the future of money transactions, are mainly implemented with blockchain technology. However, blockchain suffers from scalability issues. Sharding is the leading solution for blockchain scalability. Sharding splits the blockchain network into sub-chains called shards/committees. Each shard processes a sub-set of transactions, rather than the entire network processing all transactions. This raises security issues for sharding-based blockchain protocols. In this paper, we propose a novel methodology to analyze the security of these protocols (e.g., OmniLedger and RapidChain). In particular, this methodology estimates the failure probability of one sharding round taking into consideration the failure probabilities of all shards. To illustrate the effectiveness of the estimated failure probability, we conduct a numerical analysis of our methodology based on a huge number of trials. Finally, we compute confidence intervals to accurately estimate the failure probability and compare our methodology with existing approaches.

INDEX TERMS Blockchain, security analysis, sharding, failure probability, hypergeometric distribution.

I. INTRODUCTION

In recent years, blockchain, which is the underlying technology behind digital cryptocurrencies, e.g., Bitcoin [1] and Ethereum [2], has attracted considerable attention from both academia and industry. Blockchain plays a significant role in emerging fields such as Internet of Things (IoT), the healthcare sector, edge computing, artificial intelligence and the government sector. All these emerging fields benefit from blockchain's decentralization, immutability, robustness, security, transparency and peer-to-peer network that records digital transactions (e.g., cryptocurrency transfer). However, Blockchain has a number of open issues such as scalability [10]. Indeed, scalability is one of the key limitations and the main challenge of blockchain [13]; while traditional centralized payment systems (e.g., Visa [5]) can handle 1000s of transactions per second (tx/s), Bitcoin and Ethereum process about 7 and 15 tx/s, respectively. Several solutions, to the scalability issue, have been proposed in the literature, such as sharding (e.g., Elastico [7], OmniLedger [8], RapidChain [9]), Directed Acyclic Graph (e.g., [16]), Plasma [14], and Lightning Network [15]. The most promising solutions of the scalability in the blockchain literature make use of sharding [10]. Sharding splits/shards the blockchain into sub-chains called shards/committees. Each shard processes a sub-set of transactions rather than the entire network processing all transactions. This increases the throughput (i.e., number of transactions per second) of the network. However, sharding may compromise the blockchain security. Indeed, for the blockchain to be secure, all shards need to satisfy the committee resiliency (i.e., maximum percentage of malicious nodes that a shard can tolerate without being compromised); throughout the paper we will use the terms committee and shard interchangeably. In most networks, this resiliency is 33% (e.g., Elastico [7] and OmniLedger [8]); beyond that resiliency, a consensus instance is fundamentally insecure. The critical issue is that even if the whole network falls well under the total resiliency (i.e., maximum

The associate editor coordinating the review of this manuscript and approving it for publication was Jonghoon Kim.

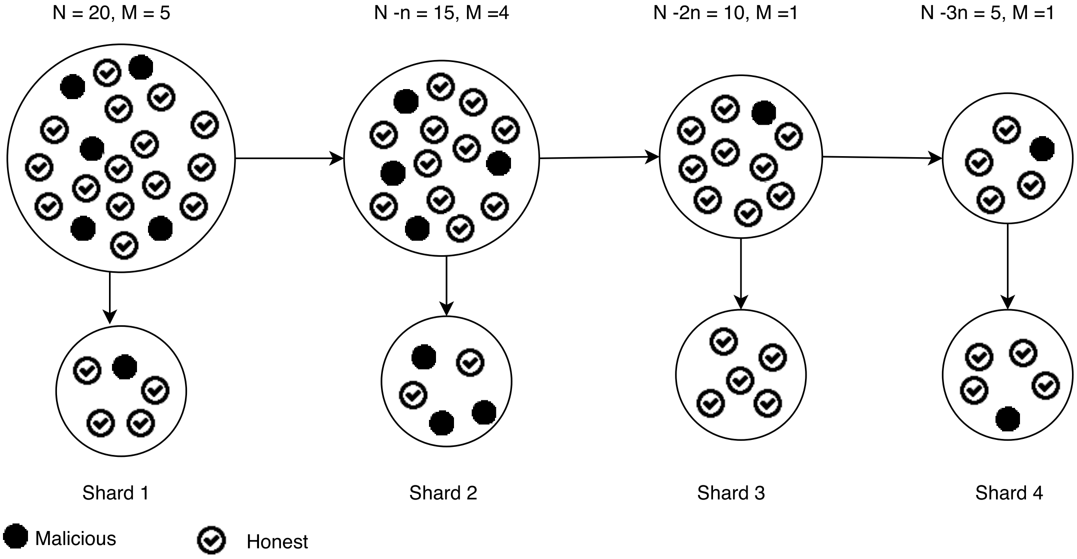

FIGURE 1. **Sharding divides the network into subsets (shards), which means only a shard can handle a set of transactions, rather than the entire network. A scenario where there is a single shard takeover attack (shard 2 in this case).**

percentage of malicious nodes that the blockchain network can tolerate without being compromised); this limit is 25% in most blockchain networks, e.g., Elastico [7] and OmniLedger [8]), a single shard could be compromised. Figure 1 shows a scenarios in which a network, that contains 20 nodes with 25% malicious nodes (i.e., 5 malicious nodes), is split evenly into 4 shards where 3 malicious nodes end up in shard 2. This means that 60% of the nodes in shard 2 are malicious, which is bigger than the committee resiliency (33%). This is known as a single shard takeover attack. In sharding-based blockchain protocols, the network is compromised if only one shard is compromised (i.e., 1% attack). In this paper, we analyze the security of sharding-based blockchain protocols. In particular, we compute the failure probability of the whole network by taking into consideration the failure probability of each committee. The key contribution of this paper is to propose a novel methodology that outperforms the computation accuracy of existing approaches [3], [4], [9]. The limitations of these approaches [3], [4], [9] come from the fact that they assume that the failure probability of the first committee is indicative of the failure probability of any other committee; more specifically, they assume that the failure probability of one epoch (i.e., fixed time period; e.g., once a day) is the failure probability of the first committee times the number of committees [3], [4], [9]. However, when the sampling, to partition the network into shards, is done without replacement, the samples are not independent; this means that when we sample the first committee, it is clear that the parameterizations of the model change (i.e., the number of nodes in the network, as well as the number of malicious nodes). Thus, the failure probability of the second committee will be different from the first, and the third will be different from the first and the second, and so on. In addition, the changes in the values of the parameters increase with the sampling process (e.g., values when sampling the second shard are very different from the values when sampling the fifth shard compared to the values when sampling the third shard). This means that the inaccuracy of the failure probability estimate proposed in [3], [4], [9], grows with the number of committees. Our methodology computes the real failure probability of each committee, then computes the failure probability of the entire network in one sharding round (aka, one epoch), taking into consideration the failure probabilities of all committees. The contributions of this paper can be summarized as follows:

- We develop a probabilistic methodology to analyze the security of sharding-based blockchain protocols. This methodology corrects and outperforms, in terms of accuracy, existing approaches;
- We estimate the failure probability and compute the confidence intervals (CIs) in order to lower and upper bound the estimated failure probability;
- We compare the proposed methodology with existing approaches;
- We identify the parameters that impact the security of sharding-based blockchain protocols (e.g., the size of the committee, the number of sharding rounds in a predefined period of time and the number of nodes in the network).

The paper is organized as follows. Section II presents definitions and notations used in the paper; in addition, it presents the details of the proposed methodology. Section III evaluates the proposed methodology. Finally, Section IV concludes the paper.

II. METHODOLOGY

In this section, we propose a methodology to estimate/compute the failure probability of one sharding round.

TABLE 1. Notations.

Notation	Description
N	Total number of nodes
n	Committee size
M	Total number of malicious nodes
m_i	Number of malicious nodes in committee i
r	Committee resiliency
R	Total resiliency
λ	Number of committees
p_e	Epoch failure probability
$h(N, M, n, i)$	Hypergeometric distribution with parameters N, M and n
$H(N, M, n, i)$	Cumulative hypergeometric distribution with parameters N, M and n
CI	Confidence interval
Y_f	Average number of years to fail
E_s	Expected number of sharding rounds until failure
N_t	Number of trials
N_{sy}	Number of sharding rounds per year
f_p	Failure probability for one sharding round
$\hat{f}_p$	Estimated failure probability for one sharding round

A. ABBREVIATIONS AND DEFINITIONS

Table 1 shows the list of symbols/variables that are used to describe the proposed approach.

Definition 1 Cumulative Hypergeometric Distribution): The cumulative hypergeometric distribution $H(N, M, n, m)$ is the sum of the probability distribution function $h(N, M, n, i)$ for all $i \geq m$, which can be expressed as follows:

$$H(N, M, n, m) = \sum_{i \geq m} h(N, M, n, i) \tag{1}$$

where

$$h(N, M, n, i) = \frac{\binom{M}{i}\binom{N-M}{n-i}}{\binom{N}{n}} \tag{2}$$

Definition 2 (Committee Resiliency: The maximum percentage of malicious nodes that the committee is able to contain whereas still being secure.

Definition 3 (Total Resiliency: The maximum percentage of malicious nodes that the whole network is able to contain whereas still being secure.

Definition 4 (Failure Probability: The probability that the number of malicious nodes exceeds the malicious nodes limit (i.e., maximum percentage of nodes that can act in a malicious manner, e.g., in case of RapidChain [9], this limit is 50% of nodes in a committee and 33% in the network) in the network/committee.

B. HYPERGEOMETRIC DISTRIBUTION

In sharding-based blockchain protocols, the process of assigning nodes to shards can be modeled as sampling without replacement because the committees do not overlap. When the sample is done without replacement, we make use of hypergeometric distribution instead of binomial distribution [4]. Indeed, assigning nodes from the network to shards can be modeled as sampling without replacement because the committees can not overlap. When the sampling is done without replacement, the hypergeometric distribution yields better approximation compared to the binomial's, especially when the sample's is bigger than 10% of the entire network [4], [12]. Let X_i denote the random variable corresponding to the number of malicious nodes in committee i and $P(X_i = m_i)$ denote the failure probability that committee i contains m_i malicious nodes.

We assume that we have a network of N nodes where M nodes ($M < N$) are malicious. The probability that a node is malicious is $p = \frac{M}{N}$. We split N nodes into committees where each committee has a size $n = \frac{N}{\lambda}$ where λ is the number of committees. When we sample the first committee, the parameterizations of the model change (i.e., N and M); in particular, N changes to $N - n$ and M changes to $M - m_1$, where m_1 is the number of malicious nodes sampled in committee 1. Then, when we sample the second committee, $N - n$ changes to $N - 2n$ and $M - m_1$ changes $M - m_1 - m_2$, where m_2 is the number of malicious nodes sampled in the committee 2. The third committee will have m_3 malicious nodes, and committee λ will have m_λ malicious nodes such that $m_1 + m_2 + \ldots + m_\lambda = M$ (see Figure 1). The distribution of the first committee can be modeled by the hypergeometric distribution with the parameters N, M and n as follows:

$$X_1 \sim \mathrm{H}(N, M, n); \tag{3}$$

Similarly, the second committee can be modeled by the hypergeometric distribution with the parameters $N - n$, $M - m_1$ and n as follows:

$$X_2 \sim \mathrm{H}(N - n, M - m_1, n); \tag{4}$$

And for the third committee we get:

$$X_3 \sim \mathrm{H}(N - 2n, M - (m_1 + m_2), n); \tag{5}$$

Finally, the distribution of committee λ (last committee) can be expressed as follows:

$$X_\lambda \sim \mathrm{H}(N - (\lambda - 1)n, M - \sum_{i=1}^{\lambda-1} m_i, n). \tag{6}$$

The probability density function of $X = (X_1, X_2, \ldots, X_\lambda)$ (i.e., joint distribution) is given in (7).

$$\begin{aligned} P(X_1 &= m_1, \ldots, X_\lambda = m_\lambda) \\ &= \mathrm{h}(N, M, n, m_1) \times \mathrm{h}(N - n, M - m_1, n, m_2) \ldots \\ &\quad \times \mathrm{h}(N - (\lambda - 1)n, M - \sum_{i=1}^{\lambda-1} m_i, n, m_\lambda) \\ &= \prod_{j=0}^{\lambda-1} h(N - jn, M - \sum_{i=0}^{j} m_i, n, m_{j+1}) \end{aligned} \tag{7}$$

$$\begin{aligned} P(X_1 &= m_1, X_2 = m_2, \ldots, X_\lambda = m_\lambda) \\ &= \frac{\binom{n}{m_1}\binom{n}{m_2} \cdots \binom{n}{m_\lambda}}{\binom{N}{M}} \\ &= \frac{\prod_{j=1}^{\lambda} \binom{n}{m_j}}{\binom{N}{M}} \end{aligned} \tag{8}$$

The distribution in (7) is difficult and complex to compute. Using **Theorem 1** (see proof in Appendix), (7) can be rewritten as (8).

Theorem 1: Let $X = \{X_1, X_2, \dots, X_\lambda\}$ be a random vector such that $X_i \sim \mathrm{H}(N-(i-1)n, M-\sum_{i=1}^{i-1} m_i, n)$ for all i in $\{1, 2, \dots, \lambda\}$. We have:

$$\prod_{j=0}^{\lambda-1} h(N-jn, M-\sum_{i=0}^{j} m_i, n, m_{j+1}) = \frac{\prod_{i=1}^{\lambda}\binom{n}{m_i}}{\binom{N}{M}} \quad (9)$$

A simple way to prove that the distribution in (7) is the distribution in (8), is as follows: We have N nodes and we need to pick M malicious nodes out of them. Thus, the total number of possibilities is $\binom{N}{M}$. For shard 1, the number of possibilities to arrange m_1 from n is $\binom{n}{m_1}$, for shard 2 is $\binom{n}{m_2}$, and for shard λ is $\binom{n}{m_\lambda}$. Consequently, the number of all possibilities taking into account all shards is the product of $\binom{n}{m_i}$ for $i \in \{1, 2, \dots, \lambda\}$. To compute the required probability, we need to divide this product by $\binom{N}{M}$.

Now, to ensure that our methodology adapts well to the situation, **Lemma 1** (see proof in Appendix) proves that the probability distribution in (8) is a proper Probability Distribution Function (PDF).

Lemma 1: The probability in (8) is a proper PDF; this means that:

$$\sum_{i=0}^{n}\sum_{j=0}^{n}\cdots\sum_{k=0}^{n} P(X_1 = m_{1i}, \dots, X_\lambda = m_{\lambda k}) = 1 \quad (10)$$

Note that m_1 malicious nodes in shard 1 can assume any of the following values: $n, n-1, \dots$ and 0. Similarly, m_2 malicious nodes in shard 2 can assume any of the following values: $n, n-1, \dots$ and 0, and so on until the last shard. Thus, the distributions in (7) and (8) represents only one particular outcome. To consider all the possible outcomes, we need to compute the joint hypergeometric distribution, which is expressed in (11).

$$\begin{aligned} &P(X_1 \le nr, X_2 \le nr, \dots, X_\lambda \le nr) \\ &= \sum_{m_1=0}^{nr}\sum_{m_2=0}^{nr}\cdots\sum_{m_\lambda=0}^{nr}\binom{n}{m_1}\binom{n}{m_2}\cdots\binom{n}{m_\lambda}\Big/\binom{N}{M} \quad (11)\end{aligned}$$

Finally, the failure probability (the probability that at least one committee fails) can be expressed as follows:

$$f_p = 1 - P(X_1 \le nr, X_2 \le nr, \dots, X_\lambda \le nr) \quad (12)$$

Even after the simplification we make (from (7) to (8)), the probability in (12) is still complex and difficult to compute, especially, when we consider a huge number of nodes. For this reason, in the section III, we estimate this probability instead of computing it.

C. EXISTING APPROACHES

In this section, we present existing approaches that are devoted to analyze the security of sharding-based blockchain protocols [3], [4], [9]. More specifically, we present Hoeffding's bound since it is the better bound (in terms of accuracy) proposed in [3], [4] as well as RapidChain methodology [9].

1) HOEFFDING's BOUND

We present Hoeffding's bound [11] in order to compare it with the proposed methodology. We choose Hoeffding's bound because it is the accurate bound, proposed in [3], [4]. This bound can be expressed as follows:

$$H(N, M, n, m) \le F(y), \quad (13)$$

where

$$F(y) = \left(\left(\frac{p}{p+y}\right)^{p+y}\left(\frac{1-p}{1-p-y}\right)^{1-p-y}\right)^{n}, \quad (14)$$

$p = \frac{M}{N}$ and $m = (p+y)n$ with $y \ge 0$.

Hence, we can bound the failure probability of one committee with resiliency r as follows:

$$H(N, M, n, nr) \le F(y), \quad (15)$$

where

$$y = r - p, \quad (p \le R).$$

Hafid *et al.* [3], [4] compute the epoch failure probability by multiplying the failure probability for one committee by the number of committees $\lambda = \frac{N}{n}$. In addition, it is possible to ignore the bootstrap probability (i.e., the probability that the committee election fails in the first epoch) since it is too small (e.g., for RapidChain [9], this probability is smaller than $2^{-26.36}$). The epoch failure probability (p_e) can thus be bounded as follows:

$$p_e \le \lambda \times F(y). \quad (16)$$

2) RapidChain METHODOLOGY

In this section, we present RapidChain methodology [9] to analyze security of sharding-based blockchain protocols. Unlike Ethereum-sharding [19] and OmniLedger [8] that use binomial distribution to analyze the security of their sharding-based blockchain protocols, RapidChain methodology uses the hypergeometric distribution. Note that using binomial distribution does not model correctly the sampling [4]. However, the limitation of RapidChain methodology comes from assuming that the failure probability of the first committee is the same as the other committees; this is because RapidChain methodology assumes that the failure probability of one epoch (i.e., one sharding round) is the failure probability of the first committee times the number of committees. As reported in Section I, the parametrizations of the model change after we sample a shard; thus, in practice each shard has its own failure probability.

The failure probability for a committee with resiliency r by using the cumulative hypergeometric distribution is expressed as follows:

$$H(N, M, n, n\,r) = \sum_{k=\lfloor nr \rfloor}^{n} \frac{\binom{M}{k}\binom{N-M}{n-k}}{\binom{N}{n}} \quad (17)$$

In RapidChain, they compute the epoch failure probability by multiplying the failure probability of the first committee by the number of committees. By ignoring the bootstrap probability, the epoch failure probability can be expressed as follows:

$$p_e = \lambda \times H(N, M, n, n\, r) = \lambda \times \sum_{k=\lfloor nr \rfloor}^{n} \frac{\binom{M}{k}\binom{N-M}{n-k}}{\binom{N}{n}} \tag{18}$$

D. COMPUTING CONFIDENCE INTERVALS

In this section, we investigate the reliability of simulations in estimating the failure probability. For this purpose, we would like to compute the confidence intervals in order to lower and upper bound the estimated failure probability. There are different and several methods to compute confidence intervals including *Normal approximation interval*, *Wilson score interval* [17], [18], *Jeffreys interval* [17], *Clopper–Pearson interval* [17], and *Agresti–Coull interval* [17]. A commonly and popular method to compute confidence intervals is *Normal approximation interval*. This method is based on the *Central Limit Theorem (CLT)*; it is inaccurate and unreliable when the sample size is small or the success probability (the failure probability in our case) is close to 0 or 1.

In this paper, we choose *Wilson score interval* since this method has been shown to be the most accurate and the most robust [17], [18]. *Agresti-Cull* method also provides a good accuracy for larger sample sizes [17].

E. YEARS TO FAIL

To measure the security of a given protocol, we propose to compute the average number of years to failure. To perform this computation, we need to determine the failure probability of epoch per sharding round, which refers to the failure probability that at least one committee fails. The average number of years to fail corresponding to the proposed methodology is given by:

$$Y_f = \frac{E_s}{N_{sy}}, \quad where\ E_s = \frac{1}{f_p} \tag{19}$$

The average number of years to fail corresponding to Hoeffding's bound as well as RapidChain methodology is given by:

$$Y_f = \frac{E_s}{N_{sy}}, \quad where\ E_s = \frac{1}{p_e} \tag{20}$$

III. RESULTS AND EVALUATION

In this section, we present a simulation-based evaluation of our methodology and we compare it with existing contributions including Hoeffding's bound [3], [4], and RapidChain methodology [9].

A. SIMULATION SETUP

To estimate the probability proposed by our methodology (i.e., the probability in (12)), we use *NumPy Python library*, which offers mathematical functions, random number generators, etc. In particular, we use *numpy.array()* to set up an array of M malicious nodes and $N - M$ honest nodes. We also use *numpy.random.choice()* to distribute **randomly** and **without replacement** these nodes across shards. Whenever, we distribute nodes without replacement across shards; we know the number of malicious nodes in each shard. If only one shard exceeds the limit (committee resiliency), we save 1 (i.e., failure), otherwise we save 0. Once this procedure is complete, we have one **trial/simulation**. To consider all the possibilities (i.e., the possible number of malicious nodes in each shard), we need to repeat this trial a large number of times. After repeating this procedure, we sum the numbers that we save (i.e., 1 or 0) and we divide by the number of trials to get the estimated failure probability. For example, let us assume we executed $N_t = 10000$ trials and we encountered at least one shard failure in each of 500 trials; in this case, the estimated failure probability is:

$$\hat{f_p} = \frac{500}{10000} = 0.05$$

The relation between the exact failure probability (f_p) and the estimated failure probability ($\hat{f_p}$) can be expressed as follows:

$$\mid f_p - \hat{f_p} \mid \xrightarrow{N_t \longrightarrow +\infty} 0 \tag{21}$$

TABLE 2. Parameter Settings.

Parameter	Value
N	1000, 2000, 3500, 4000, and 5000
r	0.333
R	0.250

Table 2 shows the values of the parameters used in the simulations. In Table 2, we assume that the number of malicious nodes in the network is the maximum number of malicious that the network can support (for Elastico and OmniLedger [7], [8], this maximum number should not exceed 25% of the entire network); this means that M assumes 25% ($M = R \times N$) of the entire network. Note that we can assume values smaller than 25% of the entire network. For the values of N, we assume different values of the network size for the purpose of analyzing how the size of the network impacts its security.

B. RESULTS AND ANALYSIS

Table 3 shows the estimated failure probability of the proposed methodology when varying the size of the committees (125, 200, 250) as well as the number of trials (10^4, 10^5, 10^6). Table 3 illustrates the Wilson score confidence interval for the purpose of computing bounds (i.e., computing lower and upper bounds) to better bound and estimate the failure probability. In addition, Wilson score confidence interval allows us to bound the failure probability with a high confidence rate of 95% and with a low error rate of 5%; this is means that, we are confident 95% that the estimated failure probability is between lower and upper bounds of Wilson score CI.

In particular, Table 3 shows that when the size of the committee increases the failure probability decreases. In addition, Table 3 shows that as the number of trials increases the width

TABLE 3. Estimated failure probability of one sharding round vs. committee size and number of trials.

Number of Trials (N_t)	Committee Size (n)	Failure Probability (Estimated)	Wilson Score Interval		
			Lower	Upper	Width
10 000	125	0.1063	0.1004	0.1124	0.0060
	200	0.0095	7.77E-03	1.15 E-02	1.91 E-03
	250	0.0017	1.06E-03	2.72E-03	8.29 E-04
100 000	125	0.10319	1.01E-01	1.05E-01	1.88 E-03
	200	0.00808	7.54E-03	8.65E-03	5.57E-04
	250	0.001	8.16E-04	1.22E-03	2.02E-04
1 000 000	125	0.105443	1.04E-01	1.06E-01	5.97E-04
	200	0.00785	7.69E-03	8.00E-03	1.57E-04
	250	0.001004	–	–	–

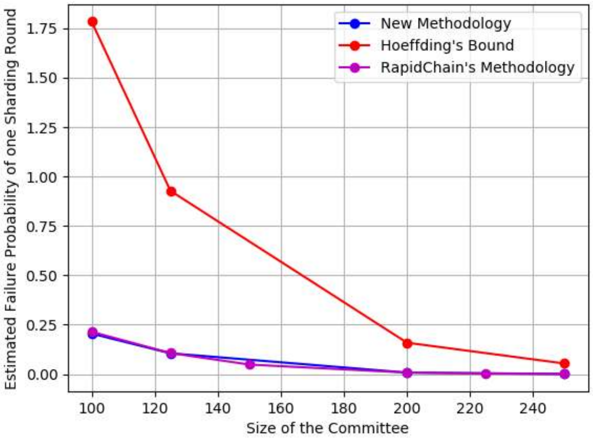

FIGURE 2. Estimated failure probability of one sharding round versus the size of the committee.

of Wilson score interval gets smaller; this means that, as the number of trials increases, we better bound (lower and upper bound) the failure probability.

It is worth noting that we could not run a very large number of trials due to the limited performance of our personal computer. A fundamental question we need to answer is ''How does the number of trials influence the estimated failure probability?''. To answer this question, we make use of confidence intervals. Table 3 shows a lower bound and an upper bound of the estimated failure probability using Wilson score confidence interval. For 1000000 number of trials computed by a regular computer (i7-2677M CPU 1.80 GHz and 6GB RAM), the execution time (running time) is 249.84 seconds, which is about 4.16 minutes. Table 3 shows that the ''width'' of Wilson score interval gets smaller as the number of trials gets larger. This means that, as the number of trials gets larger we better bound the estimated failure probability. However, when the number of trials gets larger, we need a supercomputer to calculate/estimate the failure probability in a reasonable time. It turns out that we have to make a trade-off between accuracy and computational overhead.

Figure 2 compares the estimated failure probability computed by using our methodology and that of Hoeffding's bound and RapidChain when varying the size of the committee (100-250) in a network of 1000 nodes. We observe (as expected) that the failure probability decreases as the size of the committee increases. As mentioned in section I, Hoeffding's bound and RapidChain methodology allow us to compute **"false"** failure probabilities since they estimate/compute the failure of the first shard and multiply it by the number of shards to get the epoch failure probability. Let us consider an example to show that existing approaches [3], [4], [9] produce inaccurate results. Let assume a network that contains $N = 1000$ nodes and each shard contains $n = 25$ nodes. The failure probability (by using RapidChain's methodology) for one epoch (one sharding round) is 1.569, which is bigger than 1. This means that RapidChain's methodology computes "false" probabilities. The failure probability (by using Hoeffding's bound) for one epoch is 9.118, which is bigger than 1. However, the proposed methodology computes (by considering $N_t = 100000$) 0.99987, which is smaller than 1 and it will not assume values greater than 1 because it is a proper probability distribution (see **Lemma 1**). Table 4 shows the failure probabilities, computed by the three methods, and the corresponding years to fail. It is worth noting that we did consider a small committee size (i.e., n=25) to show that existing approaches compute probabilities that are bigger than 1 (which is not correct); indeed, the smaller committee size, the bigger the failure probability. In this example, RapidChain (resp. the approaches in [3], [4]) computes a failure probability that is equal to 1.569 (resp. 9.118). The smaller the size of the committee the bigger the failure probability; thus, by decreasing the size, we can show that the failure probabilities computed by the existing approaches [3], [4], [9] exceed 1. Finally, it worth noting that as the number of years to fail decreases; this means that computing "false" probabilities impacts the number of years to fail, which impacts the security of the network. Figure 3 compares the estimated years to fail using our methodology with that of Hoeffding's bound and RapidChain's when varying the size of the committee (100-250) in a network of 1000 nodes. More specifically, Figure 3 illustrates that as the size of the committee increases the number of years to fail increases; this is expected since when the size of the committee increases, the failure probability decreases (Figure 2) leading to increasing the number of years to fail.

Figure 4 shows the number of trials (varying from 10000-1000000) versus the width of Wilson score confidence

TABLE 4. Comparison (in terms of failure probability and years to fail) between the proposed methodology and the existing approaches.

Methods	$\hat{f_p}$ [a]	Y_f
The proposed methodology	0.99987	2.74 E-03
RapidChain methodology [9]	1.56929	1.74E-03
Hoeffding's bound [3], [4]	9.11876	3.00E-04

[a]: As reported above, these probabilities are computed by considering a network of $N = 1000$ nodes and a committee size of $n = 25$. For the probability computed by the proposed methodology, we executed 100000.

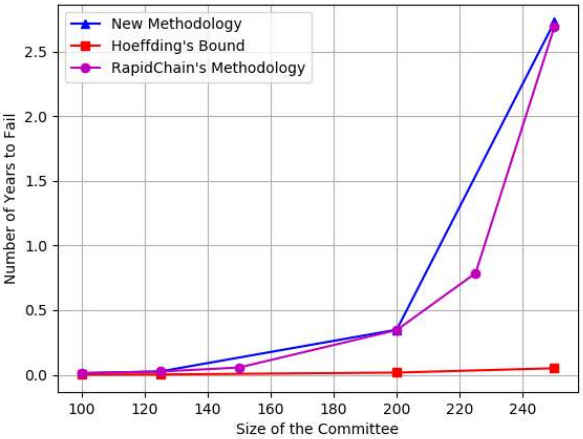

FIGURE 3. Years to fail versus the size of the committee.

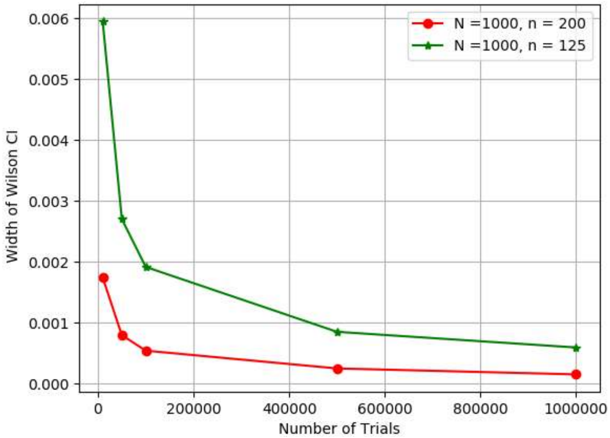

FIGURE 4. Number of trials versus the width of Wilson score confidence interval.

interval. We observe that as the number of trials increases the width of Wilson score interval gets smaller; we conclude that as the number of trials gets larger we better bound the estimated failure probability (as expected).

Figure 5 illustrates the number of trials (varying from 10000-1000000) versus the running time in seconds in a network of N = 1000 nodes. We observe that as the number of trials increases the running time ‘‘sharply'' increases due to the limited performance of our machine. From Figures 4 and 5, we conclude that as the number of trials increases we better estimate the failure probability but the running time sharply increases. It turns out that we have to make a trade-off between accuracy and computational overhead.

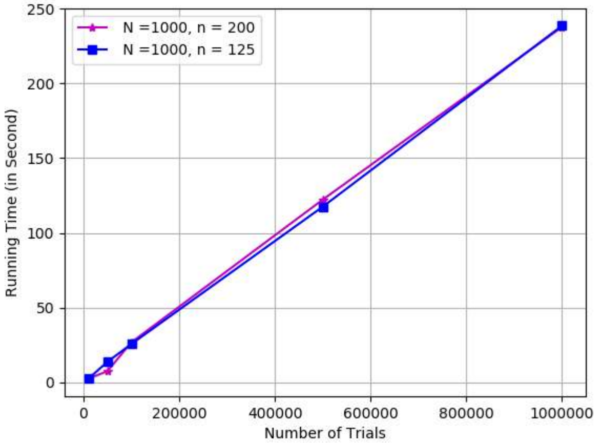

FIGURE 5. Number of trials versus the running time (in second).

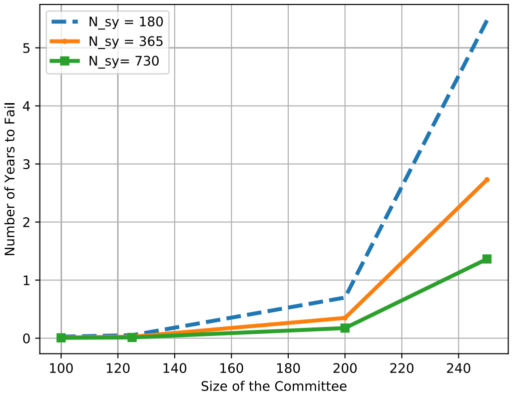

FIGURE 6. Number of years to fail for different numbers versus the size of the committee for the different number of sharding rounds per year (N_{sy}).

Figure 6 shows the number of years to fail for different numbers of sharding rounds per year ($N_{sy} = 180$, $N_{sy} = 360$, and $N_{sy} = 730$) when varying the size of the committee (100-250) in a network of $N = 1000$ nodes. We observe that as the number of sharding rounds per year decreases the number of years to fail increases; this means that, as the number of sharding rounds per year decreases the security of the network increases. We conclude that the number of sharding rounds impacts the security of sharding-based blockchain protocols.

Figure 7 shows the failure probability of one sharding round for different network's sizes (N = 1000, N = 2000, N = 4000) when varying the committee's size (100-250). Specifically, this failure probability is calculated by using the proposed methodology for $N_t = 1000000$ trials. We observe that as the network's size increases the estimated failure probability increases; this means that the size of the network affects its security.

Finally, we identify numerous factors that impact the security of the network, which are the size of the committee, the number of years to fail, and the size of the network. Now,

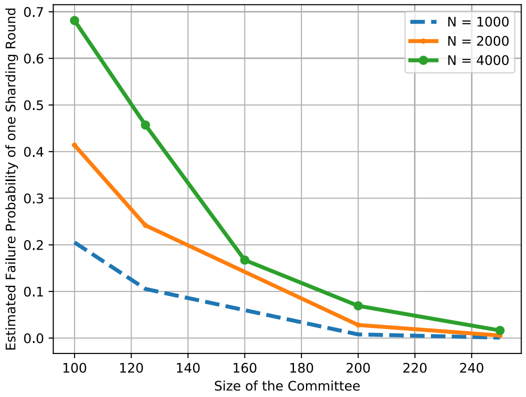

FIGURE 7. Failure probability of one sharding round versus the size of the committee (n) for the different network's sizes (N).

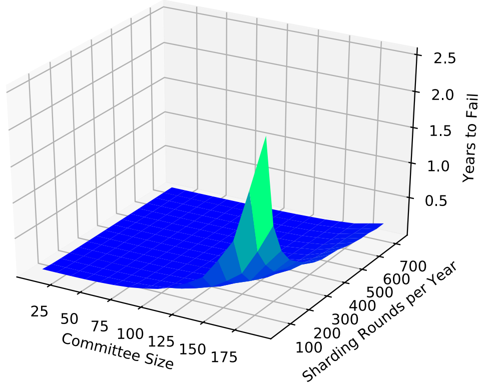

FIGURE 8. Years to fail versus the size of the committee and the number of sharding rounds per year.

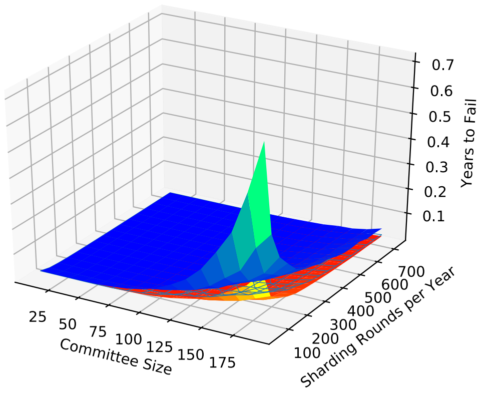

FIGURE 9. Years to fail versus the size of the committee and the number of sharding round per year for different network's sizes.

let us determine the best combination of the values of these factors to achieve the best security. In practice, the network size is given (i.e., an average) since it is public blockchain (i.e., users can leave/join at any time). However, we can increase/decrease the size of the committee and the number of sharding rounds per year in order to determine a predefined level of security (i.e., a predefined number of years to fail).

Let us consider some 3D graphs to show the best combination that gives us the best security (the bigger number years to fail). Figure 8 shows the number of years to fail versus the size of the committee (n varying from 10 to 200) and the number of sharding rounds per year (N_{sy} varying from 45 to 730) by considering $N_t = 100000$ in a network of $N = 1000$ nodes. We observe that for $n = 192.421$ and $N_{sy} = 22.589$ we have the best combination that achieves the biggest number of years to fail, which is about 2.58026 years. Figure 9 shows the years to fail for different network's sizes ($N = 2000$, $N = 3500$ and $N = 5000$) versus the size of the committee (n varying from 10 to 200) and the number of sharding rounds per year (N_{sy} varying from 45 to 730) by considering $N_t = 10000$. We observe three surfaces; the higher one corresponds to $N = 5000$ nodes, followed by the surface that corresponds to $N = 3500$ nodes, and the last one corresponds to $N = 2000$ nodes. We conclude that as the network's size increases the number of years to fail increases, which shows again that the size of the network impacts its security.

IV. CONCLUSION

In summary, the paper proposes a novel methodology to analyze and investigate the security of sharding-based blockchain protocols. This methodology corrects the existing approaches [3], [4], [9]. In particular, we estimate the failure probability of the entire network in one sharding round taking into account the failure probability of each committee/shard. To validate and confirm that our methodology gives better estimation, we compute confidence intervals using Wilson score method since it is the most accurate and robust. After estimating the failure probability, we can measure the security of the network by estimating the number of years to fail.

A. PROOF OF THEOREM 1

First we prove the equality for $\lambda = 2$.

For $\lambda = 2$, we have $M = m_1 + m_2$ and $N = 2n$. Let $A_2 = P(X_1 = m_1, X_2 = m_2)$ We need to prove that:

$$A_2 = \frac{\binom{n}{m_1}\binom{n}{m_2}}{\binom{N}{M}} \tag{22}$$

We have:

$$\begin{aligned}
A_2 &= \prod_{j=0}^{1} h(N - jn, M - \sum_{i=0}^{j} m_i, n, m_{j+1}) \\
&= \frac{\binom{M}{m_1}\binom{N-M}{n-m_1}}{\binom{N}{n}} \times \frac{\binom{M-m_1}{m_2}\binom{N-n-(M-m_1)}{n-m_2}}{\binom{N-n}{n}} \\
&= \frac{\binom{M}{m_1}\binom{N-M}{n-m_1}}{\binom{N}{n}} \times \frac{\binom{m_2}{m_2}\binom{n-m_2}{n-m_2}}{\binom{n}{n}} \\
&= \frac{\binom{M}{m_1}\binom{N-M}{n-m_1}}{\binom{N}{n}} \times a_2 \\
&= \frac{\frac{M!}{m_1!(M-m_1)!} \times \frac{(N-M)!}{(n-m_1)!(n-m_2)!}}{\frac{N!}{n!n!}} \times 1 \\
&= \frac{M!(N-M)!}{m_1!m_2!(n-m_1)!(n-m_2)!} \times \frac{n!n!}{N!}
\end{aligned}$$

$$= \frac{n!}{m_1!(n-m_1)!} \times \frac{n!}{m_2!(n-m_2)!} \times \frac{M!(N-M)!}{N!}$$
$$= \frac{\binom{n}{m_1}\binom{n}{m_2}}{\binom{N}{M}}$$

Now, let us prove the equality for $\lambda = 3$.

For $\lambda = 3$, we have $M = m_1 + m_2 + m_3$ and $N = 3n$.

And

$$A_3 = \prod_{j=0}^{2} h(N - jn, M - \sum_{i=0}^{j} m_i, n, m_{j+1})$$

We need to prove that:

$$A_3 = \frac{\binom{n}{m_1}\binom{n}{m_2}\binom{n}{m_3}}{\binom{N}{M}} \tag{23}$$

Let m_0 assumes 0,

$$A_3 = \prod_{j=0}^{2} h(N - jn, M - \sum_{i=0}^{j} m_i, n, m_{j+1})$$
$$A_3 = \prod_{j=0}^{2} h(N - jn, M - \sum_{i=0}^{j} m_i, n, m_{j+1})$$
$$= \frac{\binom{M}{m_1}\binom{N-M}{n-m_1}}{\binom{N}{n}} \times \frac{\binom{M-m_1}{m_2}\binom{N-n-(M-m_1)}{n-m_2}}{\binom{N-n}{n}} \times a_3$$
$$= \frac{\frac{M!}{m_1!(M-m_1)!} \times \frac{(N-M)!}{(n-m_1)!(N-M-(n-m_1))!}}{\frac{N!}{n!(N-n)!}}$$
$$\times \frac{\frac{(M-m_1)!}{m_2!m_3!} \times \frac{(N-n-(M-m_1))!}{(n-m_2)!(n-m_3)!}}{\frac{(N-n)!}{n!n!}} \times 1$$

By simplifying the previous expression, we get:

$$A_3 = \frac{n!}{m_1!(n-m_1)!} \times \frac{n!}{m_2!(n-m_2)!}$$
$$\times \frac{n!}{m_3!(n-m_3)!} \times \frac{M!(N-M)!}{N!}$$
$$= \frac{\binom{n}{m_1}\binom{n}{m_2}\binom{n}{m_3}}{\binom{N}{M}}$$

Finally, we get:

$$A_3 = \frac{\binom{n}{m_1}\binom{n}{m_2}\binom{n}{m_3}}{\binom{N}{M}}$$

Now, let us prove the equality for λ. Let m_0 assumes 0.

For λ, we have $M = \sum_{k=1}^{\lambda} m_i$ and $N = \lambda n$.

And

$A_\lambda = \prod_{j=0}^{\lambda-1} h(N - jn, M - \sum_{i=0}^{j} m_i, n, m_{j+1})$.

We need to prove that:

$A_\lambda = \frac{\prod_{i=1}^{\lambda}\binom{n}{m_i}}{\binom{N}{M}}$

We have:

$$A_\lambda = \prod_{j=0}^{\lambda-1} h(N - jn, M - \sum_{i=0}^{j} m_i, n, m_{j+1})$$
$$= \prod_{j=0}^{\lambda-2} h(N - jn, M - \sum_{i=0}^{j} m_i, n, m_{j+1}) \times a_\lambda$$
$$= \frac{\binom{M}{m_1}\binom{N-M}{n-m_1}}{\binom{N}{n}} \times \frac{\binom{M-m_1}{m_2}\binom{N-n-(M-m_1)}{n-m_2}}{\binom{N-n}{n}}$$
$$\times \frac{\binom{M-(m_1+m_2)}{m_3}\binom{N-2n-(M-(m_1+m_2))}{n-m_3}}{\binom{N-2n}{n}} \times \ldots$$
$$\times \frac{\binom{M-\sum_{i=0}^{k-2} m_i}{m_{k-1}}\binom{N-(k-2)n-(M-\sum_{i=0}^{k-2} m_i)}{n-m_{k-1}}}{\binom{N-(k-2)n}{n}}$$
$$\times \frac{\binom{M-\sum_{i=0}^{k-1} m_i}{m_k}\binom{N-(k-1)n-(M-\sum_{i=0}^{k-1} m_i)}{n-m_k}}{\binom{N-(k-1)n}{n}}$$
$$\times \frac{\binom{M-\sum_{i=0}^{k} m_i}{m_{k+1}}\binom{N-kn-(M-\sum_{i=0}^{k} m_i)}{n-m_{k+1}}}{\binom{N-kn}{n}} \times \ldots$$
$$\times \frac{\binom{M-\sum_{i=0}^{\lambda-3} m_i}{m_{\lambda-2}}\binom{N-(\lambda-3)n-(M-\sum_{i=0}^{\lambda-3} m_i)}{n-m_{\lambda-2}}}{\binom{N-(\lambda-3)n}{n}}$$
$$\times \frac{\binom{M-\sum_{i=0}^{\lambda-2} m_i}{m_{\lambda-1}}\binom{N-(\lambda-2)n-(M-\sum_{i=0}^{\lambda-2} m_i)}{n-m_{\lambda-1}}}{\binom{N-(\lambda-2)n}{n}} \times 1$$

By substituting each Binomial coefficient by its algebraic expression, we get:

$$A_\lambda = \frac{\frac{M!}{m_1!(M-m_1)!} \times \frac{(N-M)!}{(n-m_1)!(N-M-(n-m_1))!}}{\frac{N!}{n!(N-n)!}}$$
$$\times \frac{\frac{(M-m_1)!}{m_2!(M-(m_1+m_2))!} \times \frac{(N-n-(M-m_1))!}{(n-m_2)!(N-M-(n-(m_1+m_2)))!}}{\frac{(N-n)!}{n!(N-2n)!}}$$
$$\times \cdots$$
$$\times \frac{\frac{(M-\sum_{i=0}^{k-2} m_i)!}{m_2!(M-(\sum_{i=0}^{k-1} m_i))!} \times \frac{(N-n-(M-\sum_{i=0}^{k-2} m_i))!}{(n-m_{k-1})!(N-M-(n-\sum_{i=0}^{k-1} m_i))!}}{\frac{(N-(k-2)n)!}{n!(N-(k-1)n)!}}$$
$$\times \frac{\frac{(M-\sum_{i=0}^{k-1} m_i)!}{m_k!(M-(\sum_{i=0}^{k} m_i))!} \times \frac{(N-n-(M-\sum_{i=0}^{k-1} m_i))!}{(n-m_k)!(N-M-(n-\sum_{i=0}^{k} m_i))!}}{\frac{(N-(k-1)n)!}{n!(N-kn)!}}$$
$$\times \frac{\frac{(M-\sum_{i=0}^{k} m_i)!}{m_{k+1}!(M-(\sum_{i=0}^{k+1} m_i))!} \times \frac{(N-n-(M-\sum_{i=0}^{k} m_i))!}{(n-m_{k+1})!(N-M-(n-\sum_{i=0}^{k+1} m_i))!}}{\frac{(N-kn)!}{n!(N-(k+1)n)!}}$$
$$\times \cdots$$
$$\times \frac{\frac{(m_{\lambda-2}+m_{\lambda-1}+m_\lambda)!}{m_{\lambda-2}!(M-(m_{\lambda-1}+m_\lambda))!} \times \frac{(3n-(m_{\lambda-2}+m_{\lambda-1}+m_\lambda))!}{(n-m_{\lambda-2})!(2n-(m_{\lambda-1}+m_\lambda))!}}{\frac{(3n)!}{n!(2n)!}}$$
$$\times \frac{\frac{(m_{\lambda-1}+m_\lambda)!}{m_{\lambda-1}!m_{\lambda-1}!} \times \frac{(2n-(m_{\lambda-1}+m_\lambda))!}{(n-m_{\lambda-1})!(n-m_\lambda)!}}{\frac{(2n)!}{n!n!}} \tag{24}$$

By simplifying the previous expression, we get (25), as shown at the top of the next page.

B. PROOF OF LEMMA 1

First, let

$$A = \sum_{i=0}^{n}\sum_{j=0}^{n}\cdots\sum_{k=0}^{n} P(X_1 = m_{1i}, X_2 = m_{2j}, \ldots, X_\lambda = m_{\lambda k})$$

$$
\begin{aligned}
&= \frac{M!(N-M)!n!}{m_1!(n-m_1)!N!} \times \frac{n!}{m_2!(n-m_2)!} \times \dots \frac{n!}{m_{k-1}!(n-m_{k-1})!} \times \frac{n!}{m_k!(n-m_k)!} \times \frac{n!}{m_{k+1}!(n-m_{k+1})!} \times \dots \\
&\quad \times \frac{n!}{m_{\lambda-2}!(n-m_{\lambda-2})!} \times \frac{n!n!}{m_{\lambda-1}!m_\lambda!(n-m_{\lambda-1})!(n-m_\lambda)!} \\
&= \frac{\frac{n!}{m_1!(n-m_1)!} \times \frac{n!}{m_2!(n-m_2)!} \times \dots \times \frac{n!}{m_{k-1}!(n-m_{k-1})!} \times \dots \frac{n!}{m_{\lambda-2}!(n-m_{\lambda-2})!} \times \frac{n!}{m_{\lambda-1}!(n-m_{\lambda-1})!} \times \frac{n!}{m_\lambda!(n-m_\lambda)!}}{\frac{N!}{M!(N-M)!}} \\
&= \frac{\binom{n}{m_1}\binom{n}{m_2} \times \dots \times \binom{n}{m_{k-1}}\binom{n}{m_k}\binom{n}{m_{k+1}} \times \dots \times \binom{n}{m_{\lambda-2}}\binom{n}{m_{\lambda-1}}\binom{n}{m_\lambda}}{\binom{N}{M}} \\
&= \frac{\prod_{i=1}^{\lambda}\binom{n}{m_i}}{\binom{N}{M}} \qquad (25)
\end{aligned}
$$

where

$$m_{ij} = j, \quad \forall i \in \{1, 2, \dots, \lambda\}, \ \forall j \in \{0, 1, \dots, n\}.$$

We need to prove that the sum over this probability equals to 1. We have:

$$
\begin{aligned}
A &= \sum_{i=0}^{n}\sum_{j=0}^{n}\cdots\sum_{k=0}^{n} P(X_1 = m_{1i}, X_2 = m_{2j}, \dots, X_\lambda = m_{\lambda k}) \\
&= \sum_{i=0}^{n}\sum_{j=0}^{n}\cdots\sum_{k=0}^{n} \frac{\binom{n}{m_{1i}}\binom{n}{m_{2j}}\cdots\binom{n}{m_{\lambda k}}}{\binom{N}{M}}
\end{aligned}
$$

By using Generalized Vandermonde's identity [6], we get:

$$
\begin{aligned}
A &= \frac{\binom{n+\dots+n}{m_1+\dots+m_\lambda}}{\binom{N}{M}} \\
&= \frac{\binom{\lambda n}{M}}{\binom{N}{M}} \\
&= \frac{\binom{N}{M}}{\binom{N}{M}} \\
&= 1 \qquad (26)
\end{aligned}
$$

Finally, we have:

$$\sum_{i=0}^{n}\sum_{j=0}^{n}\cdots\sum_{k=0}^{n} P(X_1 = m_{1i}, X_2 = m_{2j}, \dots, X_\lambda = m_{\lambda k}) = 1 \qquad (27)$$

This means that P is a proper probability distribution function.

REFERENCES

[1] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, Apr. 2014. [Online]. Available: https://gavwood.com/paper.pdf

[3] A. Hafid, A. S. Hafid, and M. Samih, "A methodology for a probabilistic security analysis of sharding-based blockchain protocols," in *Proc. Int. Congr. Blockchain Appl.* Springer, 2019, pp. 101–109.

[4] A. Hafid, A. S. Hafid, and M. Samih, "New mathematical model to analyze security of sharding-based blockchain protocols," *IEEE Access*, vol. 7, pp. 185447–185457, 2019, doi: 10.1109/ACCESS.2019.2961065.

[5] *Visa*. Accessed: Mar. 23, 2020. [Online]. Available: https://usa.visa.com/run-your-business/small-business-tools/retail.html

[6] H. W. Gould, "Some generalizations of Vandermonde's convolution," *Amer. Math. Monthly*, vol. 63, no. 2, pp. 84–91, 1956.

[7] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 17–30.

[8] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "OmniLedger: A secure, scale-out, decentralized ledger via sharding," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 583–598.

[9] M. Zamani, M. Movahedi, and M. Raykova, "RapidChain: Scaling blockchain via full sharding," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Jan. 2018, pp. 931–948.

[10] A. Hafid, A. S. Hafid, and M. Samih, "Scaling blockchains: A comprehensive survey," *IEEE Access*, vol. 8, pp. 125244–125262, 2020, doi: 10.1109/access.2020.3007251.

[11] W. Hoeffding, "Probability inequalities for sums of bounded random variables," in *The Collected Works of Wassily Hoeffding*. Springer, 1994, pp. 409–426.

[12] J. Wroughton and T. Cole, "Distinguishing between binomial, hypergeometric and negative binomial distributions," *J. Statist. Edu.*, vol. 21, no. 1, pp. 1–17, Mar. 2013.

[13] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *IEEE Access*, vol. 8, pp. 16440–16455, 2020.

[14] J. Poon and V. Buterin, "Plasma: Scalable autonomous smart contracts," White paper, 2017, pp. 1–47. [Online]. Available: https://plasma.io/plasma.pdf

[15] J. Poon and T. Dryja, *The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments*, document DRAFT Version 0.5.9.2, 2016, pp. 1–59. [Online]. Available: https://lightning.network/lightning-network-paper.pdf

[16] Y. Sompolinsky, Y. Lewenberg, and A. Zohar, "SPECTRE: A fast and scalable cryptocurrency protocol," 2017, *arXiv:1710.09437*. [Online]. Available: https://arxiv.org/abs/1710.09437

[17] L. D. Brown, T. T. Cai, and A. DasGupta, "Interval estimation for a binomial proportion," *Statist. Sci.*, vol. 16, no. 2, pp. 101–117, 2001.

[18] R. G. Newcombe, "Interval estimation for the difference between independent proportions: Comparison of eleven methods," *Statist. Med.*, vol. 17, no. 8, pp. 873–890, Apr. 1998.

[19] *Ethereum Wiki*. Accessed: Jul. 23, 2020. [Online]. Available: https://eth.wiki/sharding/Sharding-FAQs

ABDELATIF HAFID (Member, IEEE) received the B.Sc. degree in mathematics and applications from the University of Moulay Ismail, Meknes, Morocco, and the M.Sc. degree in mathematical engineering from the University of Abdelmalek Essaâdi, Tangier, Morocco. He is currently pursuing the Ph.D. degree with the University of Moulay Ismail. He is also a Visiting Research Student with the University of Montreal (UdeM), Montreal, QC, Canada. His current research interests include applied probability, statistics, and blockchain.

ABDELHAKIM SENHAJI HAFID (Member, IEEE) was a Senior Research Scientist with Bell Communications Research (Bellcore), Piscataway, NJ, USA, working in the context of major research projects on the management of next generation networks. He was also an Assistant Professor with Western University (WU), Canada. He was a Research Director of the Advance Communication Engineering Center (venture established by WU, Bell Canada, and Bay Networks), Canada. He was a Researcher with CRIM, Canada. He was a Visiting Scientist with GMD-Fokus, Germany. He was a Visiting Professor with the University of Evry, France. He is currently a Full Professor with the University of Montreal. He is also the Founding Director of the Network Research Laboratory and the Montreal Blockchain Laboratory. He is also a Research Fellow with CIRRELT, Montreal, QC, Canada. He has extensive academic and industrial research experience in the area of the management and design of next generation networks. His current research interests include the IoT, fog/edge computing, blockchain, and intelligent transport systems.

MUSTAPHA SAMIH received the Ph.D. degree in fundamental and applied mathematics from the University of Montpellier, France. He is currently a Full Professor with the University of Moulay Ismail, Meknes, Morocco. His current research interests include applied probability, statistics, and blockchain.