A Tractable Probabilistic Approach to Analyze Sybil Attacks in Sharding-Based Blockchain Protocols

Abdelatif Hafid, *Member, IEEE,* Abdelhakim Senhaji Hafid, and Mustapha Samih,

Abstract—Blockchain like Bitcoin and Ethereum suffer from scalability issues. Sharding is one of the most promising and leading solutions to scale blockchain. The basic idea behind sharding is to divide the blockchain network into multiple committees, where each processing a separate set of transactions, rather than the entire network processes all transactions. In this paper, we propose a probabilistic approach to analyze the security of sharding-based blockchain protocols. Based on this approach, we investigate the threat of Sybil attacks in these protocols. The key contribution of our paper is a tractable probabilistic approach to accurately compute the failure probability that at least one committee fails and ultimately compute the probability of a successful attack. To show the effectiveness of our approach, we conduct a numerical and comparative analysis of the proposed approach with existing approaches.

Index Terms—blockchain scalability, sharding, security analysis, Sybil attacks, failure probability, hypergeometric distribution, generating function.

1 Introduction

Since the inception of Bitcoin [1], interest in blockchain technology, from Industry and Academia, has been booming. Moreover, Blockchain has been used extensively in almost all industry segments, including internet of things [11], cryptocurrencies [1], [2], education [31], the healthcare sector [22], the industrial sector [12], the financial Sector [21], and the E-Voting [23]. The inherent characteristics of blockchain provide properties like transparency, decentralization, auditability, and security. A blockchain is a distributed database that is organised as a list of ordered blocks, where the committed blocks are immutable. With all these attractive characteristics, one of the key limitation of blockchain is *scalability* [5]; indeed, the number of transactions that can be processed per second is small (e.g., up to 7 for Bitcoin [1] and 15 for Ethereum [2]). This is unacceptable for most traditional centralized payment systems that require 1000s of transactions per second (tx/s) (e.g., Visa handles an average of 1700 tx/s [13]). A number of solutions to scale blockchain have been proposed; we can classify them into two categories: (1) On-chain solutions: they propose modifications to the blockchain protocols, such as sharding (e.g., [8], [2]) and block size increase (e.g., [29]); and (2) off-chain solutions (aka layer 2 solutions): these are built on the blockchain protocols; they process certain transactions (e.g., micro-payment transactions) outside the blockchain and only record important transactions (e.g., final balances) on the blockchain. Examples of layer 2 solutions include Lightning Network [27], Raiden Network [30], and Plasma [20].

In this paper, we focus on the sharding solution. The key idea behind sharding is to divide or split the network into subsets, called shards/committees; throughout the paper, we will use the terms shard and committee interchangeably. Each shard will be working on different set of transactions rather than the entire network processing the same transactions. This idea allows the network to scale with the number of shards. It achieves high efficiency for both throughput and storage but may compromise security. Indeed, for the blockchain to be secure, all shards need to satisfy the byzantine validator limit (aka, committee resiliency, i.e., maximum percentage of malicious nodes that a committee can support). For RapidChain [8], this limit is $\frac{1}{2}$ (50%).

The key challenge is that even if the network satisfies the limit, a shard could be compromised. For example, for RapidChain [8], let us assume that the network consists of 6 shards; each shard contains 5 nodes with 33% of nodes are malicious (i.e., 10 nodes are malicious) where 1 shard has 4 of the malicious nodes and the rest evenly distributed in the other 5 shards. In this case, one shard will be 80% byzantine/malicious and thus the entire network will be insecure (because the maximum number of malicious nodes that a shard can support is 50%). This is known as a ***single shard takeover attack***, see Figure 1.

In particular, we analyze the security of sharding-based blockchain protocols by computing the failure probability using generating function. Based on this approach, we investigate Sybil attacks and identify the parameters that can deter such attacks. More specifically, we measure the security of sharding protocols by counting the number of years to fail taking into account the failure probability of each shard (i.e., the probability that a shard exceeds the committee resiliency). To do this, we make use of generating

- *A. Hafid and A. S. Hafid are with Montreal Blockchain Laboratory (mbl), Department of Computer Science and Operational Research, University of Montreal, Montreal, QC H3T 1J4, Canada. E-mail: abdelatif.hafid@umontreal.ca*
- *A. Hafid and M. Samih are with Team of EDA – Mathematical Laboratory and their Applications, Department of Mathematics, Faculty of Sciences, University of Moulay Ismail, Meknes 50050, Morocco.*

This work was supported in part by the Mohammed VI Polytechnic University - UM6P and in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

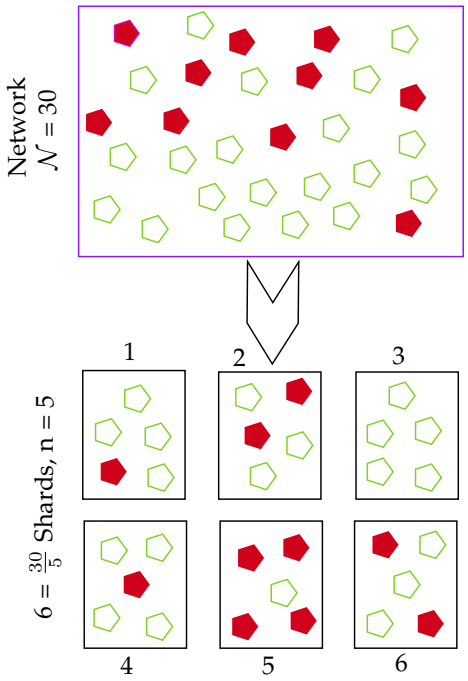

Fig. 1: Sharding divides the network into subsets (shards), which means only a shard can handles a set of transactions rather than the entire network. A scenario where there is a single shard takeover attack (shard 5 in this case).

functions. These functions are widely used in computer science. Rajan et al. [14] described many applications of generating functions in engineering and applied sciences. Kohei et al. [16] used joint generating function to analyze the retrial queues for cognitive wireless networks with sensing time. In blockchain systems, Wenjuan et al. [15] used generating function to analyze the average confirmation time of transactions. Jelena et al. [17] used probability generating function to characterize the distribution of the number of connections per node in Bitcoin network; the objective was to model the Bitcoin delivery network.

In Blockchain systems, there are several existing contributions that analyze and investigate the threat of Sybil attacks (e.g., [24], [25], and [26]), however, not in the context of sharding-based Blockchain protocols. Zhang et al. [24] introduced a new attack model that combines a double-spend attack with a Sybil attack in the Bitcoin network. The authors computed the probability of success of this new attack model and studied this attack from an economic standpoint. Iqbal et al. [25] developed a framework to explore two major security risks in the Blockchain network, namely Sybil attacks and double-spending attacks. Serena et al. [26] proposed a virtual environment to study some well-known security attacks (e.g., Sybil attacks and selfish mining) in the Bitcoin Blockchain network. In the context of sharding-based Blockchain protocols, there are a few existing works that investigate the threat of Sybil attacks (e.g., [4] and [7]).

The key contribution of this paper is to propose a tractable approach to calculate the probability that at least one committee fails using generating function and investigate Sybil attacks based on these probabilities. Note that in sharding-based blockchain protocols, the network is compromised if only one shard is compromised (i.e., 1% attack); this is why we compute the probability that at least one committee fails. The proposed approach outperforms, in terms of the computation accuracy, the mathematical models proposed in existing contributions [3], [6], [7]; it also achieves similar computation accuracy as JHDA [4] in estimating the failure probability, but with much less computational complexity.

The limitations in [3], [6], [8], [7] come from the fact that they assume that the failure probability in the first committee is indicative of the failure probability in any other committee; more specifically, they assume that the failure probability of one epoch is the failure probability of the first committee times the number of committees [3], [6], [8]. However, when the sampling is done without replacement, the samples are not independent; this means that when we sample the first committee, it is clear that the parametrizations of the model change (i.e., the number of nodes in the network which is the number of IDs, as well as the number of malicious nodes which is the number of Sybil IDs. 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, until the last committee. In addition, there are significant changes in the various parameters from the first to the second to the last sampled committee; this means that the inaccuracy of the estimate proposed in [3], [6], [7], grows with the number of committees. In a more recent work, Hafid et al. [4] filled this gap by proposing an approach, called JHDA, that takes into consideration the failure probability of each shard. However, the key limitation of this approach (i.e., JHDA) is its complexity; indeed, it is not practical to accurately compute the failure probability using JHDA; we can only estimate it by executing a large number of trials [4]. In this paper, we address this gap by proposing a new approach called Probabilistic Generating Function Approach (PGFA). Furthermore, we investigate the threat of Sybil attacks based on the probabilities of failure computed by PGFA. Specifically, the contributions of this paper can be summarized as follows:

- We propose a tractable probabilistic methodology to analyze the security of sharding-based blockchain protocols using generating function;
- We investigate the threat of Sybil attacks based on these probabilities;
- We compare, in terms of the computational complexity, PGFA with the most accurate existing approach (i.e., JHDA);
- We identify the parameters that can reduce the impact (in terms of threat severity) of Sybil attacks.

The paper is organized as follows. Section 2 presents the proposed Probabilistic Generating Function Approach (PGFA) and compares it (in terms of complexity) with JHDA. Section 3 evaluates PGFA and compares it with the Break Consensus Protocol (BCP) [7]. Finally, Section 4 concludes the paper.

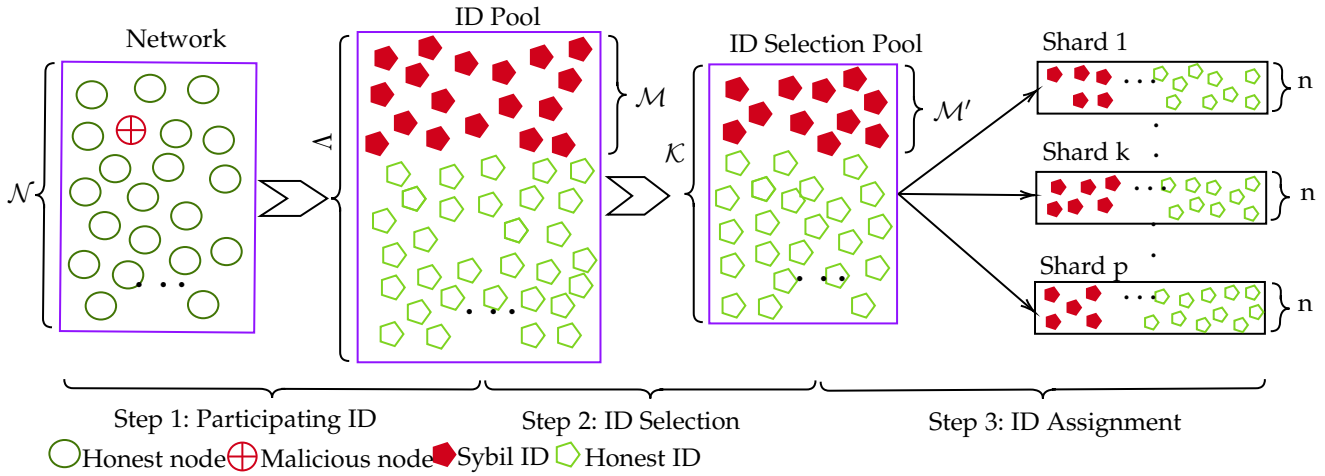

Fig. 2: Sharding-based blockchain protocol model. Step 1: Each node participates by its ID, save a malicious one, which can participate with more than one. Step 2: Using a consensus mechanism (e.g. PoW), each node competes to add its corresponding ID/IDs to the ID Selection Pool. Step 3: Random distribution of IDs from the ID Selection Pool to shards uniformly.

2 Analytical Model

In this section, we describe our approach, PGFA, to calculate/compute the probability that at least one committee fails using generating function.

2.1 Abbreviations and Definitions

Table 1 shows the list of symbols and variables that are used to describe the proposed PGFA as well as JHDA.

TABLE 1: Notations

Notation	Description
$\mathcal{N}$	Total number of nodes
Λ	Total number of IDs in the ID Pool ($\Lambda = \mathcal{N}-1+\mathcal{M}$)
$\mathcal{M}$	Total number of Sybil IDs in the ID Pool
$\mathcal{M}'$	Total number of Sybil IDs in the ID Selection Pool ($\mathcal{M}' \leq \mathcal{M}$)
m_i	Number of Sybil IDs in shard i
$\mathcal{K}$	Total number of IDs in the ID selection Pool ($\mathcal{K} \leq \Lambda$)
n	Size of the committee
r	Committee resiliency
$\mathcal{R}$	Resiliency of the ID Selection Pool
λ	Number of committees
p_e	Epoch failure probability
$\mathcal{A}$	Average number of years to failure
E_s	Expected number of sharding rounds until failure
N_s	Number of sharding rounds per year
t	Number of trials
$[x^\kappa]\,\Psi(x)$	Coefficient of x^κ in $\Psi(x)$
$\mathcal{P}$	Probability of selecting at least m Sybil IDs from the ID Pool
$\mathcal{P}'$	Probability that at least one committee fails
$\mathcal{P}''$	Probability of a successful attack

Definition 1. Committee Resiliency (r). The maximum percentage of Sybil IDs that the shard can contain whereas still being secure.

Definition 2. Total Resiliency (R). The maximum percentage of Sybil IDs that the ID Selection Pool can contain whereas still being secure. In other words, it is the maximum percentage of Sybil IDs that the ID Selection Pool can tolerate without compromising its security.

Definition 3. ID Pool. It contains honest IDs and Sybil IDs that are generated by honest nodes and Sybil nodes respectively.

Definition 4. Adversary's hash-rate. The percentage of adversary's hash-power in participating in the ID Selection Pool (e.g., if an adversary controls 30 Sybil IDs in an ID Selection Pool that contains 100 IDs, thus the adversary's hash-rate will be 30%).

Definition 5. ID Selection Pool. ID Selection Pool is constructed in each epoch using a consensus mechanism (e.g., *PoW*) and contains the total number of required and valid/qualified IDs; it contains the number of IDs that is necessary in participating in this epoch; this number depends of the security of the network.

Remark. If an honest node generates many IDs, it will not compromise the security of the network. However, if a malicious one generates many IDs (Sybil IDs), it can compromise the security of the network. Therefore, we assume in the rest of the paper that an honest node generates one ID and a malicious node can generate many Sybil IDs.

2.2 Probability of Selecting Sybil IDs from the ID Pool

In this section, we compute the probability of selecting Sybil IDs from the ID Pool during the construction of the ID Selection Pool.

Figure 2 shows a worst-case scenario of a sharding-based blockchain protocol. In this scenario, we assume that each honest node is able to generate its own ID and one strong (in terms of hash rate) malicious node (aka one adversary) generates numerous Sybil IDs. The sharding process consists of 3 steps. In the first step, each node participates by its own ID, save a malicious node, it participates with $\mathcal{M}$ IDs. In the second, we construct the ID Selection Pool from the ID Pool. More specifically, using a consensus mechanism (e.g., PoW and PoS), each node competes to generate a valid/qualified ID to join the ID Selection Pool, save the malicious, it adds $\mathcal{M}^{'}$ Sybil IDs ($\mathcal{M}^{'} \leq \mathcal{M}$) thanks to its high hash power. In the third step, we randomly distribute

(i.e., random assignment) IDs from from the ID Selection Pool to shards.

A few sharding-based Blockchain protocols (e.g. Ethereum sharding [35]) adopt the PoS-based node/ID selection method to select shard members to defend against Sybil attacks. However, in most sharding-based blockchain protocols (e.g., Elastico [10], OmniLedger [9] and RapidChain [8]), *Proof-of-Work* (*PoW*) consensus is typically used to establish the committee/shard members (committee formation) and generate the corresponding IDs; Byzantine Fault Tolerant (BFT) is used for the intra-committee consensus, which is used within a committee to create and append the blocks [8], [5]. More specifically, nodes who want to join/stay in the protocol use *PoW* consensus to generate valid IDs (i.e., public keys). Only the nodes that can solve the ID generation *PoW* puzzle (e.g. RapidChain uses a fresh fixed *PoW* puzzle [8]) can generate valid IDs. It turns out that the nodes that have a higher hash-rate have a higher probability to solve the ID generation *PoW* puzzle compared to those of lower hash-rate. The security of the network will not be compromised as long as the malicious node's computational power is limited.

Now, let $\mathcal{X}$ be a random variable that counts the number of Sybil IDs selected from the ID Pool. In the following, we compute the probability of selecting *exactly* m Sybil IDs from the ID pool (**Lemma 1**) and then the probability of selecting *at least* m Sybil IDs from the ID pool (**Lemma 2**).

Lemma 1: In a sharding-based blockchain protocol, the probability of selecting *exactly* m Sybil IDs from the ID pool is given by:

$$P(\mathcal{X} = m) = \frac{\binom{\mathcal{M}}{m}\binom{\mathcal{N}-1}{\mathcal{K}-m}}{\binom{\Lambda}{\mathcal{K}}} \tag{1}$$

Proof: In a sharding-based blockchain protocol, the process of assigning IDs to shards can be modeled as a random sampling without replacement, because the shards can not overlap. In this case, the hypergeometric distribution yields a better probability approximation compared to any other probability distribution, including that of Binomial's [18]. Thus, the proof of **Lemma 1** is a direct result from the fact that X follows a hypergeometric distribution [18], [4].

Lemma 2: In a sharding-based blockchain protocol, the probability of selecting *at least* m Sybil IDs from the ID pool can be computed as follows:

$$P(\mathcal{X} \geq m) = \sum_{s=m}^{\mathcal{K}} \frac{\binom{\mathcal{M}}{s}\binom{\mathcal{N}-1}{\mathcal{K}-s}}{\binom{\Lambda}{\mathcal{K}}} \tag{2}$$

Proof: Given the fact that $\mathcal{X}$ follows a hypergeometric distribution, the proof of **Lemma 2** can be extracted directly from the cumulative hypergeometric distribution [18].

The following subsection will be devoted to compute and calculate the probability that at least one shard fails.

2.3 Probability that at least one Shard Fails

In this section, we compute the probability that at least one shard fails in a sharding-based blockchain protocol using the proposed approach (i.e., PGFA). First, we briefly describe how JHDA [4] computes such a probability. Then, we present the details of the proposed PGFA.

2.3.1 Joint Hypergeometric Distribution Approach (JHDA)

In a more recent work, Hafid et al. [4] proposed a novel methodology-based joint hypergeometric distribution to analyze the security of sharded protocols, which can be summarized in **Theorem 1** (see proof in [4]).

Now, let $\mathcal{Y}_i$ be a random variable that computes the number of Sybil IDs in shard i. **Theorem 1** computes the probability that at least one shard fails using JHDA.

Theorem 1: In a sharding-based blockchain protocol, the probability that at least one shard fails using JHDA can be expressed as follows:

$$\mathcal{P}' = 1 - P(\mathcal{Y}_i \leq nr, i \in \{1, 2, \ldots, \lambda\}) \tag{3}$$

where

$$P(\mathcal{Y}_i \leq nr, i \in \{1, 2, \ldots, \lambda\}) \tag{4}$$

$$= \sum_{m_1=0}^{nr} \sum_{m_2=0}^{nr} \cdots \sum_{m_\lambda=0}^{nr} \prod_{i=1}^{\lambda} \binom{n}{m_i} \Big/ \binom{\mathcal{K}}{\mathcal{M}'} \tag{5}$$

The limitation of JHDA comes from the fact that this approach requires a high computation power due to its high complexity. We address this limitation in the following subsection, where we introduce a Probabilistic Generating Function Approach (PGFA) as an alternative solution to analyze the security of sharding-based blockchain protocols.

2.3.2 Generating Function Approach

Generating function transforms problems about sequences into problems about functions. With generating function, we can then apply many and several machinery problems to problems about sequences. They can also be used to find closed-form expressions for sums, to solve counting problems, and to solve recurrence relations [19]. There are few classes of generating function in common use (e.g., exponential generating function [32] and Ordinary Generating Function (OGF) [19]). In this paper, we use the ordinary class, because it is useful for solving counting problems; in particular, problems involving choosing items (i.e., sampling with/without replacement) from a set (entire network in our case). Ordinary generating function does often lead to a polynomial function by letting the coefficient of x^k be the number of ways to choose k items (nodes in our case).

2.4 Modeling

In sharding-based blockchain protocols, the process of assigning IDs to committees (or partition of the network into committees/shards) can be defined as a sampling without replacement, because the committees do not overlap [6]. When the sample is done without replacement, we make use of the hypergeometric distribution instead of the binomial distribution [6], [18]. Note that to model this sampling using the hypergeometric distribution, we need to make use of joint hypergeometric distribution to cover the failure probabilities of all committees, which is a complex/difficult to compute; it is a closed-form expression. This is the reason why we choose generating function. Generating function is fundamentally devoted to such complicated problems.

The generating function corresponding to select ***distinct*** items from a finite set (as sampling ***without replacement***) can be expressed as follows:

$$\binom{n}{0} + \binom{n}{1}x^1 + \binom{n}{2}x^2 + \binom{n}{3}x^3 + \cdots + \binom{n}{k}x^k \quad (6)$$

where $\binom{n}{k}$ is the number of ways/possibilities to choose/select k distinct items from a set of size n; it can be expressed as follows:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \quad (7)$$

The main aim of our analysis is how to divide m malicious nodes between λ committees without exceeding the resiliency of each committee. Let $m = m_1 + m_2 + \cdots + m_\lambda$ where m_1 is the number of Sybil IDs in the first committee, m_2 is the number of Sybil IDs in the second committee, and so on. We need to divide m Sybil IDs, whereas for all $i \in \{1, 2, \cdots, \lambda\}$, m_i is smaller than the resiliency of the committee (e.g., $r = \frac{1}{2}$ for RapidChain [8]).

Now, based on (6), the generating function that represents one committee can be expressed as follows:

$$\psi(x) = \binom{n}{0} + \binom{n}{1}x^1 + \binom{n}{2}x^2 + \cdots + \binom{n}{\lfloor nr \rfloor}x^{\lfloor nr \rfloor} \quad (8)$$

Precisely, if we assume (8) refers to shard 1, thus m_1 can take 0 , 1 , 2 , . . . , or $\lfloor nr \rfloor$ Sybil IDs. And if (8) refers to shard 2, m_2 also can take 0 , 1 , 2 , . . . , or $\lfloor nr \rfloor$ malicious nodes, and so on for the other shards. Based on (8), the number of possibilities for shard 1 to receive m_1 = 0 malicious nodes (i.e., shard 1 does not contain any malicious node) is $\binom{n}{0}$, the number of possibilities for shard 1 to receive m_1 = 1 malicious nodes (i.e., shard 1 contains one malicious node) is $\binom{n}{1}$; thus, the number of possibilities for shard 1 to get m_1 = $\lfloor nr \rfloor$ Sybil IDs is $\binom{n}{\lfloor nr \rfloor}$. Our first objective is to calculate the number of possibilities we can split m malicious nodes between λ committees without exceeding the committee resiliency (i.e., without exceeding the maximum number of Sybil IDs that the committee can tolerate; this number is $\lfloor nr \rfloor$ in (8)). Then, we can compute the failure probability, which represents the probability that at least one committee exceeds the committee resiliency.

Generally, the generating function for choosing elements from a union of disjoint sets is the product of the generating functions from choosing from each set [19]. In our case, we have λ committees which do not overlap (the sampling is done without replacement). Indeed, the entire network is the union of disjoint committees; it can be modeled as follows:

$$C = \bigcup_{i=1}^{\lambda} C_i \quad (9)$$

where C is a set which contains all nodes in the entire network with $card(C) = N$, C_i contains the number of nodes in committee i with $card(C_i) = n$ for all $i \in \{1, 2, \cdots, \lambda\}$, and for all $i, j \in \{1, 2, \cdots, \lambda\}$, $C_i \cap C_j = \emptyset$ for $i \neq j$. To compute the total number of ways we can distribute the m Sybil IDs across all committees, we multiply this generating function with itself λ times:

$$\Psi(x) = \left(\binom{n}{0} + \binom{n}{1}x^1 + \binom{n}{2}x^2 + \cdots + \binom{n}{\lfloor nr \rfloor}x^{\lfloor nr \rfloor}\right)^{\lambda} \quad (10)$$

where

$$\Psi(x) = (\psi(x))^{\lambda} \quad (11)$$

Now, we need to extract the coefficient of x^m in (10), which corresponds to the number of possibilities we can split m Sybil IDs across λ committees without exceeding the resiliency of each committee (note that all the committees have the same resiliency). To compute the sequence of coefficients from this generating function, we need to compute the *Taylor series* for this generating function [19]. Therefore, the required coefficient can be expressed as follows:

$$[x^m]\,\Psi(x) = \frac{\Psi^m(0)}{m!} \quad (12)$$

where $\Psi^m(0)$ is the mth derivative of $\Psi(x)$ evaluated at $x = 0$.

The required coefficient can be determined explicitly (analytically) by using (12); however, this process involves tedious and complex calculations. Instead, we can determine the coefficient computationally by using the *SymPy* package [33]; it uses the symbolic math system making the process easier to execute.

The probability that at least one committee fails is the probability that at least one committee contains more than nr Sybil IDs. This means that the number of Sybil IDs in the committee exceeds the committee resiliency. Therefore, the probability that at least one committee fails using PGFA is expressed in **Theorem 2**.

Theorem 2: In a sharding-based blockchain protocol, the failure probability that at least one committee fails using PGFA can be expressed as follows:

$$\mathcal{P}' = 1 - \frac{[x^m]\,\Psi(x)}{\binom{\mathcal{N}}{m}} \quad (13)$$

where $\binom{\mathcal{N}}{m}$ is the total number of possibilities to select m Sybil IDs from $\mathcal{N}$ IDs.

2.5 Probability of a Successful Attack

In this section, we compute the probability of a successful attack (the failure probability of the entire network); this means that we take into consideration the probability of selecting Sybil IDs from the ID Pool as well as the probability of at least one shard takeover attack.

Theorem 3: Given a sharding-based blockchain protocol, the probability of a successful attack (assumed by an adversary) can be computed as follows:

$$\mathcal{P}'' = \sum_{s=m}^{\mathcal{K}} \frac{\binom{\mathcal{M}}{s}\binom{\mathcal{N}-1}{\mathcal{K}-s}}{\binom{\mathcal{N}-1+\mathcal{M}}{\mathcal{K}}} \left(1 - \frac{[x^m]\,\Psi(x)}{\binom{\mathcal{N}}{m}}\right). \quad (14)$$

Proof: By taking into consideration both probabilities (i.e., the probability of selecting Sybil IDs from the ID pool as well as the probability that at least one shard fails), and based on **Lemma 2** and **Theorem 2**, the probability of a successful attack ($\mathcal{P}''$) can be expressed as follows.

$$\begin{aligned}\mathcal{P}^{''} &= P(\mathcal{X}=nr)\mathcal{P}^{'} + \cdots + P(\mathcal{X}=\mathcal{K})\mathcal{P}^{'} \\ &= \big(P(\mathcal{X}=nr) + \cdots + P(\mathcal{X}=\mathcal{K})\big)\mathcal{P}^{'} \\ &= \sum_{k=nr}^{\mathcal{K}} P(\mathcal{X}\geq k)\mathcal{P}^{'}\end{aligned} \tag{15}$$

2.6 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 PGFA is given by:

$$\mathcal{A} = \frac{E_s}{N_s}, \quad where \quad E_s = \frac{1}{\mathcal{P}^{''}} \tag{16}$$

The average number of years to fail corresponding to JHDA ($\mathcal{P}'$ must be calculated by **Theorem 1**) is given by:

$$\mathcal{A} = \frac{E_s}{N_s}, \quad where \quad E_s = \frac{1}{p_e} \tag{17}$$

where

$$p_e = \mathcal{P} \times \mathcal{P}'$$

2.7 Complexity Comparison

Table 2 shows a comparison (in terms of complexity) between PGFA and JHDA.

TABLE 2: Complexity Comparison between PGFA and JHDA.

Approach	Complexity (Worst case)
JHDA [4]	$O((nr)^{\lambda})$
PGFA	$O(nr)$

Table 2 shows that the computational complexity of JHDA is polynomial because we have λ-nested loops (see **Theorem 1**) while the proposed approach claims a linear complexity (see Algorithm 1, there is only one loop). We conclude that JHDA is very difficult and impractical to compute whereas PGFA shows a linear computational complexity. Thus, the feasibility of PGFA to analyze the security of sharding-based Blockchain protocols.

3 Results and Evaluation

In this section, we compare (in terms of accuracy) PGFA and JHDA [4] via simulations. In particular, we compute the probability of selecting Sybil IDs from the ID Pool to construct the ID Selection Pool as well as the failure probability that at least one committee fails using PGFA and JHDA. These probabilities represent the failure probability of the whole network in one sharding round (i.e., one epoch). We also investigate the threat of Sybil attacks and identify the parameters that impact (in terms of threat severity) these attacks.

3.1 Simulation Setup

To implement our approach, we use *SymPy Python library*, which makes the use of symbolic mathematics easier [33], (e.g., polynomial functions). Particularly, we use *sympy.Symbol()* and *sympy.Poly()* to declare the variable "x" explicitly and to explicitly express the polynomial in (8). To implement JHDA, we refer the readers to [4].

Algorithm 1: Algorithm for Computing $\mathcal{P}^{'}$ (PGFA)

Input: $\mathcal{K}$ (IDs in the ID Selection Pool), n (Committee size), $\mathcal{M}'$ (Sybil IDs), r (Committee resiliency)
Output: $\mathcal{P}^{'}$ `// Probability that at least one shard fails using PGFA`

Function PGFA($\mathcal{K}$, n, $\mathcal{M}'$, r):
- $\lambda \leftarrow \lfloor \frac{\mathcal{K}}{n} \rfloor$ `// Number of shards`
- $a \leftarrow [\ ]$ `// Empty list`
- **for** $\forall i \in \mathcal{D} = \langle \lfloor r \times n \rfloor, \lfloor r \times n \rfloor - 1, \ldots, 0 \rangle$ **do**
  - $a \leftarrow \langle \binom{n}{\lfloor r\times n \rfloor}, \binom{n}{\lfloor r\times n \rfloor - 1}, \ldots, 0 \rangle$ `/* Vector with` $\lfloor r \times n \rfloor + 1$ `components */`
  - $X \leftarrow \langle x, x, \ldots, x \rangle$ `/* Vector with` $\lfloor r \times n \rfloor + 1$ `components */`
  - $\psi(x) \leftarrow a \cdot X = \sum_{i=\lfloor r\times n \rfloor}^{0} \binom{n}{i} x$ `/*` $a \cdot X$ `means the dot product of vectors` a `and` X`.` $\psi(x)$ `is a polynomial with variable` x `and coefficients` $\{\binom{n}{i}, \forall i \in \mathcal{D}\}$ `*/`
- **end**
- $\Psi(x) \leftarrow \big(\psi(x)\big)^{\lambda} = \big(\sum_{i=}^{0} \binom{n}{i} x\big)^{\lambda}$ `/*` $\psi(x)$ `power the number of shards (`λ`) */`
- $\zeta \leftarrow [x^{\mathcal{M}'}]\Psi(x)$ `/* Coefficient of` $x^{\mathcal{M}'}$ `in` $\Psi(x)$`; it corresponds to the number of possibilities to distribute` $\mathcal{M}'$ `Sybil IDs across` λ `shards (see Equation 12) */`
- $\mathcal{P}' \leftarrow 1 - \frac{\zeta}{\binom{\mathcal{K}}{\mathcal{M}'}}$
- **return** $\mathcal{P}'$

Table 3 shows the values of the parameters used in the simulations.

TABLE 3: Parameter Settings

Parameter	Value
$\mathcal{N}$	1000, 1200, 1400
$\mathcal{K}$	400, 600, 800
$\mathcal{M}$	200, 250
$\mathcal{M}'$	125, 200, 250
r	0.25, 0.27, 0.333
$\mathcal{R}$	0.20, 0.25, 0.333
N_s	365, 185
n	80, 100, 200

3.2 Results and Analysis

Figure 3 shows a comparison between JHDA and PGFA for different ID Selection Pool sizes ($\mathcal{K}$), values of resiliency (r

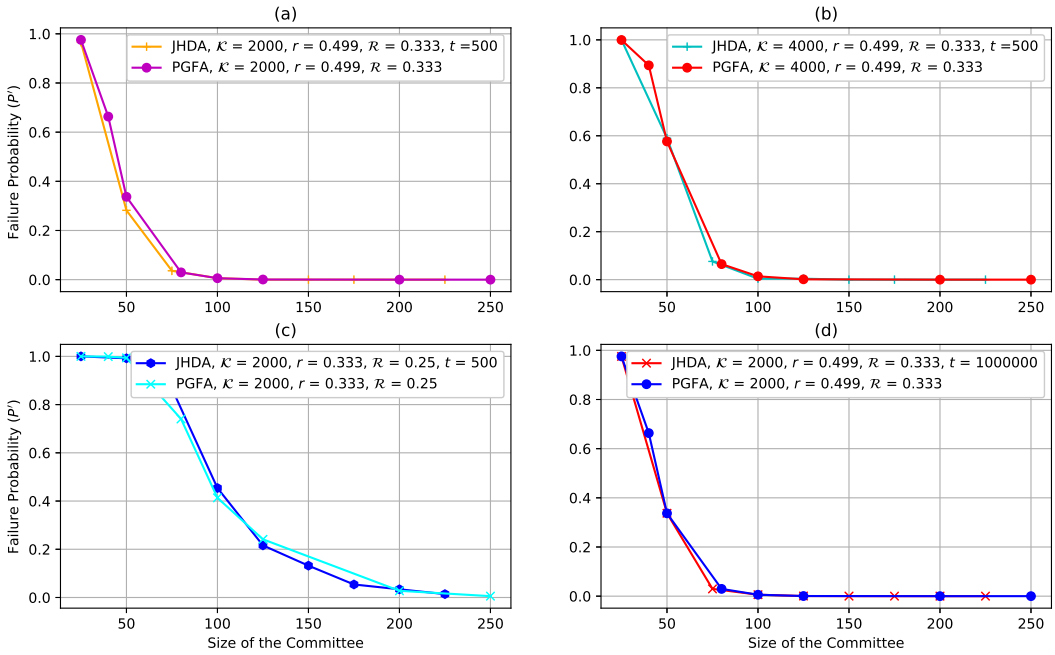

Fig. 3: Comparison between JHDA and PGFA for different network sizes, values of resiliency, and number of trials.

and $\mathcal{R}$), and numbers of trials (t) when varying the size of the committee from 25 to 250 by a step of 25.

More specifically, Figures 3(a), 3(b), and 3(c) show that the performance of FPGA is close to JHDA. Figure 3(d) shows a much closer match between the results achieved by PGFA and JHDA. This is expected since, for JHDA, when the number of trials (t) increases the estimated failure probability moves towards the exact failure probability [4]. We conclude that the proposed PGFA allows for an accurate computation of the failure probability.

Figure 4 shows the failure probability of selecting Sybil IDs from the ID Pool ($\mathcal{P}$), the failure probability for at least one shard takeover attack ($\mathcal{P}'$), and the probability of a successful attack ($\mathcal{P}''$) when varying the number of Sybil IDs (we assume that $\mathcal{M}$ = $\mathcal{M}'$; the worst case that can happen) from 10 to 200 (by a step of 10 IDs) for different network sizes, values of committee resiliency, values of ID Selection Pool resiliency, ID Selection Pool sizes, and for different values of committee size (n). These results show that the failure probability increases with the number of Sybil IDs means a high hash-rate of the adversary; this increases the probability to compromise the security of the network. In particular, Figures 4 (a), 4 (b), and 4 (c) show the failure probability of selecting Sybil IDs from the ID Pool for different network size ($\mathcal{N}$ = 1000, $\mathcal{N}$ = 1200, and $\mathcal{N}$ = 1400), ID Selection Pool resiliency ($\mathcal{R}$ = 0.333, $\mathcal{R}$ = 0.25, and $\mathcal{R}$ = 0.20), and for different ID Selection Pool size ($\mathcal{K}$ = 400, $\mathcal{K}$ = 600, and $\mathcal{K}$ = 800), respectively. More specifically, Figure 4 (a) shows that as the network size increases (in this case, we set of the size of the ID Pool Selection to 800 IDs, and the resiliency of ID Selection Pool to 0.333) the failure probability ($\mathcal{P}$) delays to assume big values; this shows up the impact of the network size on the security of the network. Figure 4 (b) shows that when the committee resiliency gets larger the failure probability ($\mathcal{P}$) delays to assume values close to 1; this shows that a higher ID Selection Pool resiliency allows for higher security.

Figure 4 (c) shows the impact of the size of the ID Selection Pool on the failure probability ($\mathcal{P}$). We observe that as the size of ID selection pool gets larger the failure probability is slow to assume values closer to 1; thus, the larger the size of the ID Selection Pool the higher the security of the network.

In addition, Figures 4 (d), 4 (e), and 4 (f) show the failure probability that at least one shard takeover attack ($\mathcal{P}'$) for different ID Selection Pool sizes, values of the committee resiliency, and committee sizes, respectively.

More specifically, Figure 4 (d) shows the impact of the size of the ID Selection Pool ($\mathcal{K}$ = 600 and $\mathcal{K}$ = 700; in this case, we set $\mathcal{N}$ to 1000, r to 0.333, and n to 100) on the failure probability ($\mathcal{P}'$). We observe that as the size of ID Selection Pool gets larger the failure probability is slow to assume values closer to 1; thus, the larger the size of the ID Selection Pool the higher the security of the network.

Figure 4 (e) shows the impact of the committee resiliency (r) on the failure probability ($\mathcal{P}'$). We observe that as the committee resiliency gets larger the failure probability is slow to assume values closer to 1; thus, the larger committee resiliency the higher the security of the network.

Figure 4 (f) shows the impact of the committee size (r) on the failure probability ($\mathcal{P}'$). We observe that as the committee size gets larger the failure probability is slow to assume large values (i.e., values closer to 1); thus, the larger the size of the committee the higher the security of the network.

Figures 4 (g), 4 (h), and 4 (i) show the probability of a successful attack ($\mathcal{P}''$) when varying the number of Sybil IDs ($\mathcal{M}$) from 10 to 200 by a step of 10. We observe that the failure probability ($\mathcal{P}''$) increases with the number of Sybil IDs (generated by an adversary). This is expected, since the

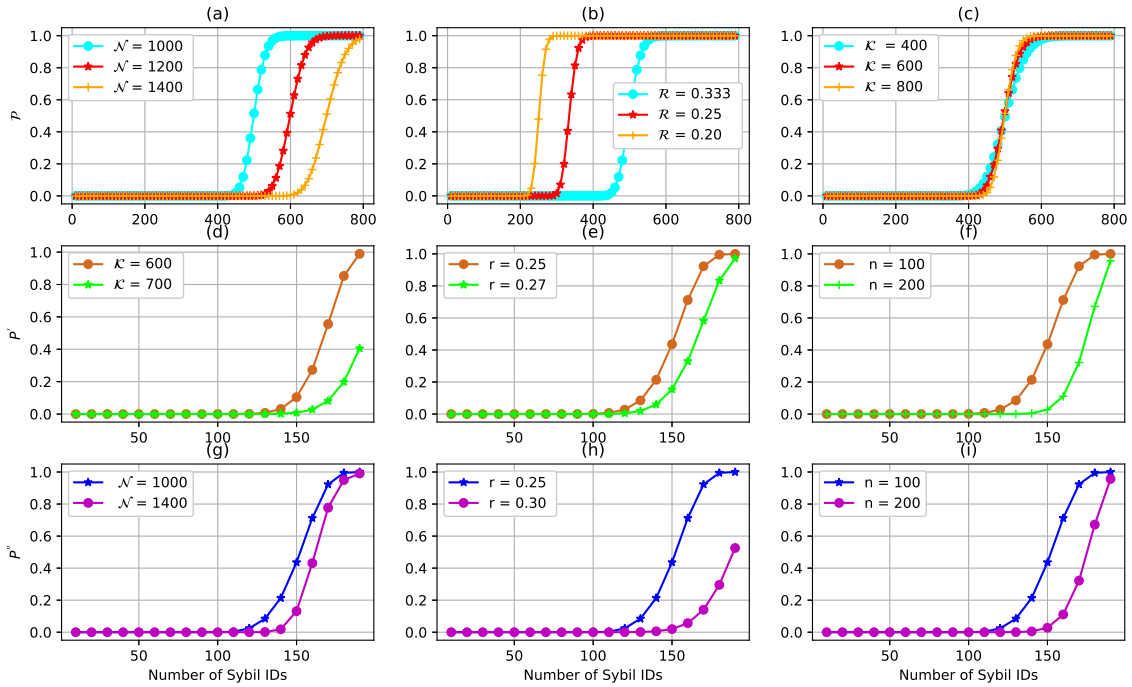

Fig. 4: (a), (b), and (c) performance of PGFA on computing failure probability of selecting Sybil IDs from the ID Pool ($\mathcal{P}$) when varying the network size ($\mathcal{N}$), values of ID Selection Pool resiliency ($\mathcal{R}$), and ID Selection Pool sizes ($\mathcal{K}$). (d), (e), and (f) failure probability for at least one shard takeover attack ($\mathcal{P}'$) when varying $\mathcal{K}$, values of the committee resiliency (r), and for different values of the size of the committee (n). (g), (h), and (i) probability of a successful attack ($\mathcal{P}''$) when varying $\mathcal{N}$, r, and n.

increase in the number of Sybil IDs leads to an increase of the chance of an adversary to compromise the security of the network.

More specifically, Figure 4 (g) shows the impact of the network size ($\mathcal{N}$) on the probability of a successful attack ($\mathcal{P}''$); in this case, we set r to 0.25, $\mathcal{R}$ to 0.10 , $\mathcal{K}$ to 800, and n to 100. We observe that as the network size gets larger the probability of a successful attack is slow to assume values closer to 1; thus, the larger size of the network the higher the security of the network.

Figure 4 (h) shows the impact of the committee resiliency (r) on the probability of a successful attack ($\mathcal{P}''$); in this case, we set $\mathcal{N}$ to 1000, $\mathcal{R}$ to 0.10, $\mathcal{K}$ to 800, and n to 100. We observe that as the committee resiliency gets larger the probability of a successful attack is slow to assume values closer to 1; thus, the larger committee resiliency the higher security of the network.

Finally, Figure 4 (i) shows the impact of the size of the committee (n) on the probability of a successful attack ($\mathcal{P}''$); in this case, we set $\mathcal{N}$ to 1000, $\mathcal{R}$ to 0.10, $\mathcal{K}$ to 800, and r to 0.25. We observe that as the size of the committee gets larger the probability of a successful attack is slow to assume values closer to 1; thus, the larger size of the committee the higher security of the network.

To sum up, Figure 4 shows that by taking advantage of PGFA we get a promising results. In particular, we identify the parameters that impact the probability of a successful attack. This means $\mathcal{N}$, $\mathcal{K}$, $\mathcal{M}$, $\mathcal{M}'$, n, λ, r, and $\mathcal{R}$.

Table 4 shows the probability of a successful attack ($\mathcal{P}^{''}$) and the average number of years to fail ($\mathcal{A}$) for different parameters (i.e., $\mathcal{N}$, $\mathcal{K}$, $\mathcal{M}$, $\mathcal{M}'$, n, λ, r, $\mathcal{R}$, and N_s). We observe that when we change the values of some parameters (even small changes), the network security can be considerably impacted. For example, Table 4 shows that for $\mathcal{R}$ = 0.2, the number of years to fail is 8623.61 and for $\mathcal{R}$ = 0.15, the number of years to fail is 1.02e-02. Indeed, a network that fails 1000s years on average has a high acceptable level of security (i.e., secure) whereas a network that fails less than one year on average is not secure enough.

Now, to validate the feasibility of our approach, we perform a comparison with another existing approach called Break Consensus Protocol attack (BCP) [7]. Rajab et al. [7] computed (by using BCP) $\mathcal{P}$, but failed to accurately compute $\mathcal{P}'$. Thus, they failed to accurately compute $\mathcal{P}''$. BCP proposed a similar model to that of Figure 2; for that reason, we compare the proposed PGFA with BCP.

Figure 5 shows the probability of a successful attack when varying the number of Sybil IDs from 10 to 200 (by a step of 10 IDs) for both PGFA and BCP approach. We observe that the probability of a successful attack computed by BCP exceeds 1. This means that BCP computes false probabilities and its formula (*Theorem 1* in [7]) does not corresponds to a proper probability distribution. In particular, they computed the probability that at least one shard fails by assuming that the probability of the first shard is indicative to the probability of the other shards; more specifically, they multiply the failure probability of the first shard by the number of shards to get the probability that at least one shard fails.

Note that the proposed probabilistic model can be adopted to different scenarios by adjusting some parameters (e.g. $\mathcal{K}$, $\mathcal{M}^{'}$). For instance, (1) A network with many

TABLE 4: Computation of Years to Fail by PGFA

$\mathcal{N}$	$\mathcal{K}$	$\mathcal{M}$	$\mathcal{M}'$	n	λ [a]	r	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{P}'$	$\mathcal{P}''$	N_s	$\mathcal{A}$ [c]	Secure
1000	800	200	200	100	8	0.333	0.20	2.04e-06	1.56e-01	3.18e-07	365	8623.61	✓
1000	800	200	200	100	8	0.333	0.20	2.04e-06	1.56e-01	3.18e-07	185	17014.16	✓
1400	800	200	200	200	8	0.333	0.20	9.25e-13	1.56e-01	1.49e-13	365	19e08	✓
1000	800	200	200	100	8	0.333	0.15	9.83e-01	1.56e-01	1.53e-01	365	1.02e-02	×
1000	800	250	250	100	8	0.333	0.20	4.79e-01	9.94e-01	4.77e-01	365	3.37e-02	×
1000	800	250	250	100	8	0.333	0.20	4.79e-01	9.94e-01	4.77e-01	185	1.13e-02	×
1000	800	250	125	100	8	0.333	0.20	4.79e-01	5.69e-06	2.73e-06	185	1980.16	✓
1000	800	250	125	80	10	0.333	0.20	4.79e-01	1.61e-04	7.73e-05	185	69.97	×

[a] $\lambda = \frac{\mathcal{K}}{n}$; [c]: Years to fail, which is calculated by using the formula described in (16).

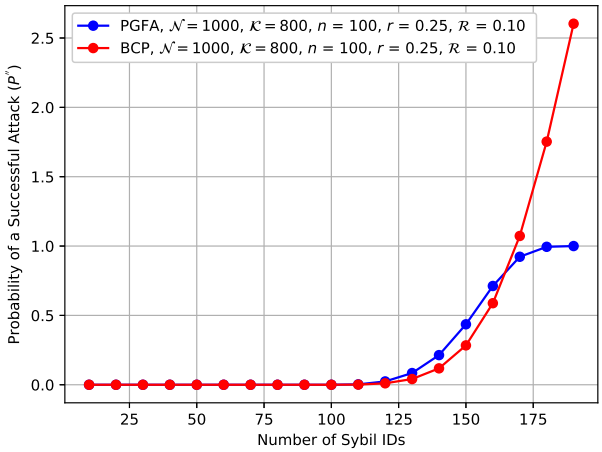

Fig. 5: Comparison between PGFA and BCP

malicious nodes where each generates numerous Sybil IDs; and (2) Honest node/nodes participate with many IDs.

There are numerous parameters that can impact the security of shading based Blockchain protocols. We can classify these parameters into two classes: Fixed parameters and adjustable parameters. Fixed parameters cannot be modified once the model is built. Examples of these parameters include shard resiliency r and ID selection pool resiliency $\mathcal{R}$; these parameters are intrinsic to the blockchain protocol and depend mainly on the type of consensus mechanisms that are used. Adjustable parameters, namely the committee size and the number of sharding rounds per year, can be set periodically (e.g., in each epoch). Our proposal allows to adjust these parameters based on the size of the network N, ID selection pool resiliency $\mathcal{R}$, shard resiliency r, and the desired level of security (i.e., the desired number of years to fail). For example, our proposal recommends a committee size of 200 and 365 sharding rounds per year (see Table 4) for a network that has 1400 nodes, ID selection pool resiliency of 0.20, and shard resiliency of 0.333, for a desired failure probability of 1.49e-13 (i.e., 19e08 years to fail).

4 Conclusion

In this paper, we propose a tractable probabilistic approach to analyze the security of sharding-based Blockchain protocols by using generating function. The proposed PGFA takes into consideration the failure probability of each committee to analyze the security instead of assuming that the failure probability for the first committee is indicative to the failure probabilities of the other committees. More specifically, we compute the failure probability that at least one committee fails. Finally, after calculating the correct failure probability (i.e., the failure probability that at least one committee fails), we propose to quantify the security of the network by computing the average number of years to failure. Furthermore, based on these probabilities, we investigate the threat of Sybil attacks. We conclude that the proposed PGFA is a promising solution to evaluate the security of existing sharding-based blockchain protocols; indeed, to the best of our knowledge, it is the most accurate and tractable approach to compute the failure probability.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Working Paper, 2008, [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, 2014, [Online] Available: https://gavwood.com/paper.pdf

[3] A. Hafid, A. S. Hafid, M. Samih, "A Methodology for a Probabilistic Security Analysis of Sharding-Based Blockchain Protocols," in *Proceedings of the International Congress on Blockchain and Applications*, Springer, 2019, pp. 101–109.

[4] A. Hafid, A. S. Hafid and M. Samih, "A Novel Methodology-Based Joint Hypergeometric Distribution to Analyze the Security of Sharded Blockchains," in *IEEE Access*, vol. 8, pp. 179389-179399, 2020, doi: 10.1109/ACCESS.2020.3027952.

[5] A. Hafid, A. S. Hafid and M. Samih, "Scaling Blockchains: A Comprehensive Survey," in *IEEE Access*, vol. 8, pp. 125244-125262, 2020, doi: 10.1109/ACCESS.2020.3007251.

[6] A. Hafid, A. S. Hafid and M. Samih, "New Mathematical Model to Analyze Security of Sharding-Based Blockchain Protocols," in *IEEE Access*, vol. 7, pp. 185447-185457, 2019, doi: 10.1109/ACCESS.2019.2961065.

[7] T. Rajab, M. M. Manshaei, M. Jadliwala, R. Murtuza and M. A. Rahman, "On the Feasibility of Sybil Attacks in Shard-Based Permissionless Blockchains," in *arXiv preprint arXiv:2002.06531*, 2020.

[8] M. Zamani, M. Movhedi, and M. Raykova, "Rapidchain: Scaling blockchain via full sharding," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2018, pp. 931–948.

[9] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, " Omniledger: A secure, scale-out, decentralized ledger via sharding," in *Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2018, pp. 583–598.

[10] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxana, " A secure sharding protocol for open blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2016, pp. 17–30.

[11] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli and M. H. Rehmani, "Applications of Blockchains in the Internet of Things: A Comprehensive Survey," *in IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1676-1717, Secondquarter 2019, doi: 10.1109/COMST.2018.2886932.

[12] D. Mazzei, G. Baldi, F. Giacomo, M. Gualtiero, P.Gabriele, R. Antonio, R. Laura, L. Rizzello, "A Blockchain Tokenizer for Industrial IOT trustless applications," *in Future Generation Computer Systems*, vol. 105, pp. 432–445, Elsevier, 2020.

[13] Visa, Accessed on: Mar. 23, 2020, [Online] Available: https://usa.visa.com/

[14] R. Chattamvelli, R. Shanmugam,"Generating Functions in Engineering and the Applied Sciences," Morgan & Claypool, 2019.

[15] W. Zhao, S. Jin, W. Yue,"Analysis of the Average Confirmation Time of Transactions in a Blockchain System," in *Proceedings of the International Conference on Queueing Theory and Network Applications*, Springer, 2019, pp. 379–388.

[16] K. Akutsu, P. Kohei, T. Phung-Duc,"Analysis of Retrial Queues for Cognitive Wireless Networks with Sensing Time of Secondary Users,"in *Proceedings of the International Conference on Queueing Theory and Network Applications*, Springer, 2019, pp. 77–91.

[17] J. Misic, V. Misic, x. Chan, M. Xiaolin, S.G. Motlagh, and Z. M. Ali,"Analysis of Retrial Queues for Cognitive Wireless Networks with Sensing Time of Secondary Users,"in *IEEE Transactions on Network Science and Engineering*, IEEE, 2019.

[18] J. Wroughton and T. Cole,"Distinguishing between binomial, hypergeometric and negative binomial distributions,"in *J. Statist. Educ.*, vol. 21, no. 1, 2013.

[19] E. Lehman, F T. Leighton, and A. R Meyer, "Mathematics for computer science," *MIT*, 2010.

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

[21] S. B. Patel, P. Bhattacharya, S. Tanwar and N. Kumar, "KiRTi: A Blockchain-based Credit Recommender System for Financial Institutions," *in IEEE Transactions on Network Science and Engineering*, doi: 10.1109/TNSE.2020.3005678.

[22] M. H. Kassab, J. DeFranco, T. Malas, P. Laplante, g. destefanis and V. V. Graciano Neto, "Exploring Research in Blockchain for Healthcare and a Roadmap for the Future," *in IEEE Transactions on Emerging Topics in Computing*, doi: 10.1109/TETC.2019.2936881.

[23] L. PANIZO ALONSO, M. GASCÓ, D. Y. MARCOS del BLANCO, J. Á. Hermida Alonso, J. Barrat and H. Aláiz Moreton, "E-Voting System Evaluation Based on The Council of Europe Recommendations: Helios Voting," *in IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 1, pp. 161-173, 1 Jan.-March 2021, doi: 10.1109/TETC.2018.2881891.

[24] S. Zhang, J-H. Lee, "Double-spending with a sybil attack in the bitcoin decentralized network," *in IEEE transactions on Industrial Informatics*, vol. 15, no. 10, pp. 5715–5722, 2019.

[25] M. Iqbal, R. Matulevičius, "Exploring sybil and double-spending risks in blockchain systems," *in IEEE Access*, vol. 9, pp. 76153–76177, 2021.

[26] L. Serena, G. D'Angelo, and S, Ferretti, "Security analysis of distributed ledgers and blockchains through agent-based simulation," *in Simulation Modelling Practice and Theory*, vol. 114, pp. 102413, 2022.

[27] J. Poon, and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," DRAFT Version 0.5.9.2, pp. 1–59, 2016, [Online] Available: https://lightning.network/lightning-network-paper.pdf

[28] H-W. Wang, " Ethereum sharding: Overview and finality," 2017, Accessed on: Sep. 8, 2019, [Online] Available: https://medium.com/@icebearhww

[29] J. Garzik, "Block size increase to 2MB," Bitcoin Improvement Proposal, Vol. 102, 2015.

[30] Raiden Network-Fast, " cheap, scalable token transfers for Ethereum," 2018, [Online] Available: https://raiden.network/

[31] A. Alammary, S. Alhazmi, M. Almasri, S. Gillani, "Blockchain-based applications in education: A systematic review," *in Applied Sciences*, vol. 9, no. 12, pp. 2400, 2019.

[32] T K. Petersen , "Inquiry-Based Enumerative Combinatorics: One, Two, Skip a Few... Ninety-Nine, One Hundred," *Springer*, 2019.

[33] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, M. Sergiu, S. Jason K, and Others, " SymPy: symbolic computing in Python," *PeerJ Computer Science*, Vol. 3, pp. e103, 2017.

[34] E. Bressert, " SciPy and NumPy: An Overview for Developers," *O'Reilly Media*, 2012.

[35] V. Buterin, "Ethereum sharding," 2017, [Online] Available: https://eth.wiki/sharding/Sharding-FAQs

Abdelatif Hafid is Postdoctoral Researcher at the University of Montreal. He is member of Montreal Blockchain Lab. He received Ph.D. degree in Applied Probability and Blockchain from the University of Moulay Ismail, Meknes, Morocco, in cooperation with the University of Montreal, Montreal, Canada. His current research interests include Applied Probability, Statistics, Machine Learning, and Blockchain.

Abdelhakim Senhaji Hafid is Full Professor at the University of Montreal. He is the founding director of Network Research Lab and Montreal Blockchain Lab. He is research fellow at CIRRELT, Montreal, Canada. Prior to joining U. of Montreal, he spent several years, as senior research scientist, at Bell Communications Research (Bellcore), NJ, US working in the context of major research projects on the management of next generation networks. Dr. Hafid was also Assistant Professor at Western University (WU), Canada, Research director of Advance Communication Engineering Center (venture established by WU, Bell Canada and Bay Networks), Canada, researcher at CRIM, Canada, visiting scientist at GMD-Fokus, Germany and visiting professor at University of Evry, France. Dr. Hafid has extensive academic and industrial research experience in the area of the management and design of next generation networks. His current research interests include IoT, Fog/edge computing, blockchain, and intelligent transport systems.

Mustapha Samih is currently a Full Professor at the University of Moulay Ismail, Meknes, Morocco. He received the Ph.D. degree in Fundamental and Applied Mathematics from the University of Montpellier, France. His current research interests include Applied Probability, Statistics, and Blockchain.