Contract-based Approach for Security Deposit in Blockchain Networks with Shards

Jing Li*, Tingting Liu†, Dusit Niyato‡, Ping Wang§, Jun Li† and Zhu Han*
*Department of Electrical and Computer Engineering
University of Houston, Houston, Texas 77004-4005, USA
Email: jli84@uh.edu, zhan2@uh.edu
†School of Electronic and Optical Engineering
Nanjing University of Science and Technology, Nanjing, 210094, China
Email: liutt@njit.edu.cn, jun.li@njust.edu.cn
‡School of Computer Engineering
Nanyang Technological University, 639798, Singapore
Email: dniyato@ntu.edu.sg
§Department of Electrical Engineering and Computer Science
York University, Toronto ON M3J 1P3 Canada
Email: pingw@yorku.ca

***Abstract*—As a decentralized ledger technology, blockchain is considered to be a potential solution for applications with highly concentrated management mechanism. However, most of the existing blockchain networks are employed with the hash-puzzle-solving consensus protocol, known as proof-of-work. The competition of solving the puzzle introduces high latency, which directly leads to a long transaction-processing time. One solution of this dilemma is to establish a blockchain network with shards. In this paper, we focus on the blockchain network with shards and adopt the security-deposit based consensus protocol, studying the problem of how to balance the security incentive and the economic incentive. Also, the inherent features of the blockchain, i.e., anonymity and decentralization, introduce the information asymmetric issue between the beacon chain and the participants. The contract theory is utilized to formulate the problem between them. As such, the optimal rewards related to the different types of validators can be obtained, as well as the reasonable deposits accordingly. Compared with the fixed deposits, the flexible deposits can provide enough economic incentive for the participants without losing the security incentives. Besides, the simulation results demonstrate that the contract theory approach is capable of maximizing the beacon chain's utility and satisfying the incentive compatibility and individual rationality of the participants.**

***Index Terms*—contract theory; security deposit; blockchain; sharding;**

I. Introduction

Blockchain was first proposed by Nakamoto in the Bitcoin project [1]. With the characteristics of decentralization, persistency, anonymity and auditability [2], it attracts significant attentions from both academia and industry, and has been applied in many fields such as finance, transportation, Internet of Things (IoT) and so on [3]. Some other well-known platforms based on blockchain technology have been developed to date, e.g. Ethereum [4].

As a collection of various technologies, one of the most important concerns of blockchain is how to coordinate the permissionless participants. Also, as a public distributed ledger, the blockchain has the problems of synchronization of transactions and the double-spending attacks. With all the considerations mentioned above, Nakamoto proposed a consensus protocol in [1], known as the Proof of Work (PoW). This protocol requires participants to solve a hash puzzle based on SHA-256 by brute-force. For each participant, it has to increase the hashrate to obtain an opportunity of winning the puzzle solving game. That means the computation power consumed by all the participants is tremendous, causing a huge waste of resources. To provide relief for this dilemma, Proof of Stake (PoS) is firstly proposed in the Bitcoin Forum [5], i.e., the leader selection relies on the number of bitcoins rather than the consumption of computational resources. Iddo *et al.* [6] point out the rationale behind the PoS that entities with stake are more suitable to retain the security in order to prevent losses caused by the system erodes. Based on the core idea of PoS, numerous frameworks with different puzzle designs have been proposed [7].

Despite being recognized as a promising approach to transaction management for permissionless networks, most existing blockchain schemes suffer from the problems of high latency and low throughput. Take the Ethereum as an example, its 1.0 version can only process 7-15 transactions per second [8]. Thus, more and more research is focused on achieving low latency and high throughput. One straightforward approach is to partition the entities into parallel sub-groups (i.e., committees), which are responsible for sustaining the sub-blocks (i.e., sharding). Sharding is a technology derived from the distributed database and can be applied to the blockchain network for realizing the scalability [9].

To further satisfy the requirements in the blockchain network with shards, protocols that support high-throughput such as [10] and [11] are proposed. Vitalik and Virgil proposed a PoS-based finality consensus protocol, i.e., Casper [12], which can be deployed atop any *proposal mechanism*. It recruits

978-1-7281-4693-5/19/$31.00 ©2019 IEEE
DOI 10.1109/Blockchain.2019.00019

a number of validators to finalize the blocks that published by the committees. As a security-deposit based economic consensus protocol [13], it requires all validators to submit a certain amount of deposits if they intend to join the network. The security feature is deemed to derive from the size of deposits rather than the number of validators. Ethereum 2.0 adopts the combination of sharding and Casper to handle the scalability issue. It sets the deposit to a fixed value, i.e., 32 ETH (A kind of token issued by Ethereum) [14], and claims that the fixed deposit that greatly exceeds gain can provide a sufficient security incentive [12]. This means the validators are forced to behave in a legal way. Otherwise, their deposits will be slashed. However, it will impair the economic incentive for those validators, whose gains are far less than 32 ETH. For those validators whose stake values are far more than 32 ETH, the security incentive constraint is weakened. Therefore, it is necessary to design an appropriate economic incentive scheme, so that it can accordingly adjust the deposit values for different validators.

Before designing such an incentive scheme, we analyze problems in the blockchain model. First of all, for a blockchain network, all participants need to be rewarded, otherwise the decentralized network cannot be sustained without any economic incentive. Second, the stake values and performance are not uniformly distributed among these participants, so it is difficult to determine the rewards, let alone the deposits. Last but not the least, for a blockchain network with shards, even though there is a beacon chain responsible for recording the administrative transactions, the problem of *information asymmetry* still exists. Owing to the anonymity of a blockchain network and weak leadership of the beacon chain, the problem mainly manifested in two aspects: (a) The beacon chain cannot determine the exact stake value of a validator, and (b) the real computation resource owned by validators are unknown to others except themselves. These aspects lead to the difficulty in designing an economic incentive mechanism.

In this paper, with the consideration of all the above problems, we are inspired by some incentive designs in the different industrial scenarios, i.e., [15] and [16], and utilize the contract theory [17] to determine the validators' rewards and ensure that the high-type validators are rewarded more than the low-type ones. The proposed contract theory based approach can balance the security incentive and economic incentive for the beacon chain and and the validators. Let the beacon chain with weak leadership be responsible for designing the contracts. The validators are classified into a variety of types according to their stake value and performance. As a result, the deposit can be set flexibly according to the different rewards. From the security incentive standpoint, deposit exceeding the rewards is guaranteed, and from the economic incentive standpoint, the deposit set according to validators' stake value is necessary.

The rest of this paper is organized as follows. In Section II, we present the system model and the utility functions of the beacon chain and validators. In Section III, we provide the specific design of contract, including problem formulation and optimal solutions. In Section IV, we illustrate the simulation results and the analysis. In Section V, we give the conclusion for the proposed scheme.

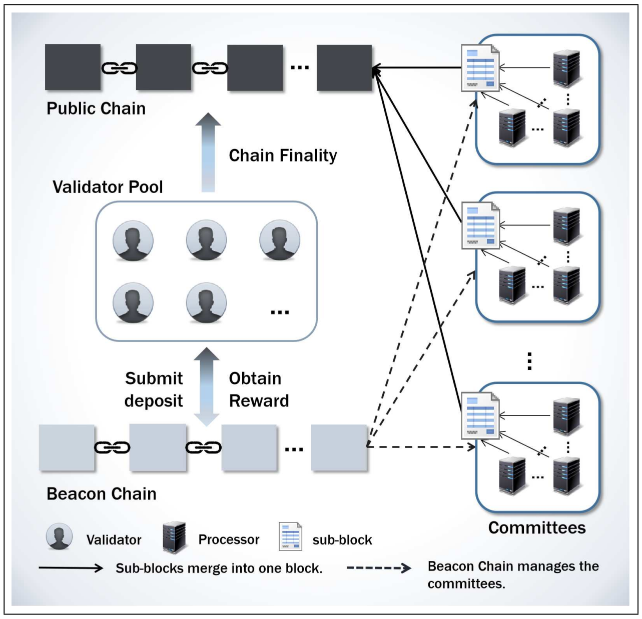

Fig. 1. The beacon chain manages the committees and recruits the validators to finalize the public chain.

II. System Model

Fig. 1 illustrates a blockchain network with shards, which is supported by the Casper [12] protocol. Generally, it consists of one public chain, one beacon chain, and many committees. The public chain is constituted by verified sub-blocks with all the finalized transactions, the beacon chain contains the administrative transactions that are associated with all the participants and committees, and the committees are responsible for generating sub-blocks (i.e., shardings).

Since the PoS consensus protocol is always combined with some other protocols to support a blockchain network, here we consider only the Capser as the upper layer protocol. In the Capser, the validators are recruited at the beginning of each *epoch*, all the validators who intend to join the network have to submit a certain amount of deposit. Their task is to finalize the blocks for the current blockchain network by generating and broadcasting the vote messages. The voting time is related to the validator's performance, as the validator with high performance takes less time on generating the vote message.

In this section, we first classify the validators into different types according to their performance and stakes, and adopt the contract theory framework $(R_v(T_v^{-1}), T_v^{-1})$ with time (T_v^{-1})-reward (R_v) bundle, where T_v denotes the total time caused by validators during voting, and $R_v(T_v^{-1})$ is a strictly increasing reward evaluation function over T_v^{-1}. That means a validator with high stakes and low voting latency will be classified as high types, and thus obtains more reward than that of a lower type one.

A. Validator Voting Time

According to the voting process described in Casper, the voting process mainly contains three parts: (a) the unfinalized blocks information transmission to validators, (b) the vote message generation, and (c) the vote message broadcasting and comparison. The second part requires a validator to perform the hash function and invoke the signature scheme, which will consume the validators' resource. The total time cost by a validator is the measurement of its performance. Thus, for validator v, according to the procedures mentioned above, the total time is given by:

$$T_v(r_v, S_{vote}) = \frac{B_e}{t_v^e} + \frac{C(B_e)}{r_v} + \psi S_{vote}|\mathbf{M}|, \tag{1}$$

where B_e is the size of blockchain information within the current *epoch*, $C(B_e)$ is the computation evaluation function of B_e, t_v^b is the transmission rate from the Beacon Chain to validators, and r_v is the resource that validator v can provide. Also, we adopt $\psi S_{vote}|\mathbf{M}|$ to denote the voting message broadcasting time [18], where ψ is pre-defined by analyzing the behaviors of previous validators, S_{vote} is the size of voting message, $\mathbf{M}$ is the set of validators, and $|\mathbf{M}|$ denotes the total number of validators.

The Casper sets a limit for the time cost. The voting time of a validator exceeding the limitation will be considered to be doing-nothing. According to the analysis of previous validators, we can obtain the time cost distribution of all types of validators.

B. Validator Utility Model

Casper is one of the PoS consensus protocols, wherein security is highly related to the deposit, and validator's deposit correlates with its stake value. Intuitively, we classify the would-be validators into different types according to their stake values along with their performances. We consider that validators with higher stakes are more preferred by the blockchain network and they can receive more reward. Set N types for all the would-be validators, and θ_i denotes the type, expressed as follows

$$0 < \theta_1 < ... < \theta_i < ... < \theta_N, \forall i \in \{1, ...N\}. \tag{2}$$

Moreover, one point that is different from the other protocols is that Casper allows the system to penalize the validators of doing-nothing. In other words, any validator who responses beyond the time limitation will be penalized. Let t_{max} denote the maximum acceptable time of validators' voting, and t_{end} denote the *epoch* end time.

For type-i validator based on contract (R_i, T_i^{-1}), let T_i denote the expected value of the valid time for type-i validators, the utility function is defined as

$$U_v(i) = \theta_i \nu(R_i) - \omega T_i^{-1} - \phi(t_i^*), \tag{3}$$

where $\nu(R_i)$ is a monotonically increasing function for evaluating the reward of the type-i validators, with $\nu(0) = 0$, $\nu' > 0$ and $\nu'' \le 0$, ω represents the unit resource cost of the compete voting process. To simplify the computation, we use $\phi(t_i^*)$ to evaluate the penalty for the type-i validators, and t_i^* is the expected time of doing-nothing for the type-i validators. We adopt $\theta_i \nu(R_i)$ to evaluate the rewards that is obtained from the beacon chain.

According to the Casper, the validators response under the time limitation will not be fined, i.e., for $\forall t_i^* \in \{0, ..., t_{max}\}$, we have $\phi(t_i^*) = 0$. However, if the validators do not finish their job on time, they will be fined, i.e., for $\forall \ t_i^* \in \{t_{max}, ..., t_{end}\}$, $\phi(t_i^*)$ is a monotonically increasing function for evaluating the fine of the type-i validators. We set $\phi(t_i^*)$ to be a linear function:

$$\phi(t_i^*) = \begin{cases} 0 & 0 \le t_i^* \le t_{max}, \\ \frac{\rho(R_i)}{t_{end} - t_{max}}(t_i^* - t_{max}) & t_{max} < t_i^* \le t_{end}, \end{cases} \tag{4}$$

where $\rho(R_i)$ is a monotonically increasing function regarding reward R_i, with $0 \le \rho' < \nu'$ and $\nu'' \le \rho'' \le 0$, which can be considered as the deposit of the type-i validators. Since the penalty is deducted from the deposit, so we have

$$\max \quad \phi(t_i^*) = \rho(R_i). \tag{5}$$

In fact, on the premise of the security deposit protocol, most of the doing-nothing is caused by network failure. Therefore, we assume that the probability of the network failure is uniform, which means for $\forall i \in \{1, ..., N\}$ and $i \neq j$, we have $t_i^* = t_j^*$. Set $(t_{end} - t_{max})^{-1}(t_i^* - t_{max}) = \varepsilon$, and the objective of type-i validators is to maximize the utility obtained from the Beacon Chain, described by

$$\max_{(R_i, T_i^{-1})} \quad U_v(i) = \theta_i \nu(R_i) - \omega T_i^{-1} - \varepsilon \rho(R_i). \tag{6}$$

C. Beacon Chain Model

In a blockchain network with shards, the beacon chain records all the administrative information of the committees within the network. So each of the block managers of the beacon chain can be considered as a temporary leader and responsible for designing the contracts for all types of the validators. The profit obtained from a validator v belonging to type-i is defined as

$$U_{BC}(i) = \sigma(T_i, \theta_i, R_i) - \mu R_i, \tag{7}$$

where we set

$$\begin{aligned} \sigma(T_i, \theta_i, R_i) =& a_1 \Big(\frac{\theta_i}{\bar{\theta}}\Big)^{\alpha} + a_2 \left(\sum_{j=1}^{i} \theta_j^{-2} - d \right) \Big(\frac{\rho(R_i)}{\rho(\bar{R})}\Big)^{\beta} \\ & - a_3 \Big(\frac{1}{T_{max} T_i^{-1}}\Big)^{\gamma}, \end{aligned} \tag{8}$$

and $\sigma(T_i, \theta_i, R_i) > \sigma(T_{max}, \theta, R)$. Also, $\bar{\theta}, \bar{R}$ represent the type and the related deposit in the traditional scheme, respectively. Only when the beacon chain gains more than that of traditional scheme can the utility function be considered as reasonable. For description purpose, we set

$$t_0 = \frac{T_{max}}{a_3}\Big(a_1\Big(\frac{\theta_i}{\bar{\theta}}\Big)^{\alpha} + a_2(\sum_{j=1}^{i}\theta_j^{-2} - d)\Big(\frac{\rho(R_i)}{\rho(\bar{R})}\Big)^{\beta} - a_1 - a_2(\sum_{j=1}^{i}\theta_j^{-2} - d) + a_3\Big)^{\frac{1}{r}}. \tag{9}$$

where a_1, a_2 and a_3 are pre-defined coefficients that weigh the stake type, the related deposits and response efficiency, and $\alpha > 0$, $0 < \beta < 1$ and $0 < \gamma$ are set to define the variation tendency, and μ is unit cost of the beacon chain. We have $T_i < t_0$, from (7), we can see that the gross profit includes not only the validators' voting time but also the penalties and validators' stake values.

Let λ_i denote the prior distribution probability of type i. According to all types of validators, $\forall i \in \{1, ...N\}$, the objective of the Beacon Chain is to maximize the expected utility function, which means higher ranked validators with quicker responses are more desired, expressed as

$$\max_{(R_i, T_i^{-1})} \quad U_{BC} = \sum_{i=1}^{\mathbb{N}} \lambda_i \left(a_1\Big(\frac{\theta_i}{\bar{\theta}}\Big)^{\alpha} + a_2\left(\sum_{j=1}^{i}\theta_j^{-2} - d\right)\Big(\frac{\rho(R_i)}{\rho(\bar{R})}\Big)^{\beta} - a_3\Big(\frac{1}{T_{max}T_i^{-1}}\Big)^{\gamma} - \mu R_i\right). \tag{10}$$

III. Contract Design

In this section, we formulate the problem between the beacon chain and the validators, and present the proof of individual rationality and incentive compatibility constraints. Finally, we obtain the optimal results by the convex optimization tools.

A. Problem Formulation

Here are two necessary principles for all types of validators involved in contract theory [17]: Individual Rationality (**IR**) and Incentive Compatibility (**IC**). IR means that only when a positive utility assigned by the beacon chain, can the validator accept the contract, i.e.,

$$\theta_i\nu(R_i) - \omega T_i^{-1} - \varepsilon\rho(R_i) \geq 0, \quad \forall i \in \{1, ..., N\}. \tag{11}$$

IC refers to that a validator of type-i can only obtain the maximum profit by choosing the contract (R_i, T_i^{-1}) rather than all the other contracts (R_j, T_j^{-1}), set

$$\mathcal{A}_i(R_i, T_i^{-1}) = \theta_i\nu(R_i) - \omega T_i^{-1} - \varepsilon\rho(R_i), \tag{12}$$

$$\mathcal{A}_i(R_j, T_j^{-1}) = \theta_i\nu(R_j) - \omega T_j^{-1} - \varepsilon\rho(R_j), \tag{13}$$

and thus, according to (12) and (13), IC can be expressed by

$$\mathcal{A}_i(R_i, T_i^{-1}) \geq \mathcal{A}_i(R_j, T_j^{-1}), \quad \forall i, j \in \{1, ..., N\}, i \neq j. \tag{14}$$

Finally, $\forall i \in \{1, ...N\}$, the optimization problem can be described as

$$\max_{(R_i, T_i^{-1})} \quad U_{BC} = \sum_{i=1}^{\mathbb{N}} \lambda_i \Big(a_1\Big(\frac{\theta_i}{\bar{\theta}}\Big)^{\alpha} + a_2(\sum_{j=1}^{i}\theta_j^{-2} - d)\Big(\frac{\rho(R_i)}{\rho(\bar{R})}\Big)^{\beta} - a_3\Big(\frac{1}{T_{max}T_i^{-1}}\Big)^{\gamma} - \mu R_i\Big), \tag{15}$$

s.t.

$$\begin{aligned}
&(a) \quad \mathcal{A}_i(R_i, T_i^{-1}) \geq 0, \\
&(b) \quad \mathcal{A}_i(R_i, T_i^{-1}) \geq \mathcal{A}_i(R_{i'}, T_{i'}^{-1}), i \neq i', \\
&(c) \quad 0 < \theta_1 < ... < \theta_i < ... < \theta_N, \\
&(d) \quad \max\{T_i\} \leq \min\{t_{max}, t_0\},
\end{aligned}$$

wherein (a) and (b) are the IR and IC constraints, respectively, (c) is the monotonicity condition, and (d) is the maximum value constraints. Obviously, this problem is not a convex optimization problem; however, we can find a solution in the following subsections.

B. Optimal Contract Solution

In order to solve the problem (15) formulated in Subsection (III-A), we first reduce the number of the constraints by proving the definition of Spence-Mirrless single-crossing Property, and then prove the sufficiency and the necessity of the monotonicity, and show the proofs of IC and IR constraints. The details are listed in the following steps.

Lemma 1: (Spence-Mirrlees single-crossing Property) For any contract (R_i, T_i^{-1}) with different type θ_i, if the utility function of validators satisfies the inequation as follows:

$$\frac{\partial}{\partial\theta}\Big[-\frac{\partial U/\partial R}{\partial U/\partial T^{-1}}\Big] > 0, \tag{16}$$

the number of constraints can be effectively reduced.

Proof 1: From equation (3), we can obtain

$$\frac{\partial}{\partial\theta}\Big[-\frac{\partial U/\partial R}{\partial U/\partial T^{-1}}\Big] = \frac{\nu'}{\omega} > 0. \tag{17}$$

Obviously, the result satisfies the Spence-Mirrlees single-crossing property. Thus, we can reduce the constraints and continue the proof based on this definition.

■

Lemma 2: (Monotonicity) For any contract (R_i, T_i^{-1}), $R_i \geq R_j$ and $T_i^{-1} \geq T_j^{-1}$ if and only if $\theta_i \geq \theta_j$.

Proof 2: According to the IC constraints of different types of validators, we can obtain

$$\mathcal{A}_i(R_i, T_i^{-1}) \geq \mathcal{A}_i(R_j, T_j^{-1}), \tag{18}$$

$$\mathcal{A}_j(R_j, T_j^{-1}) \geq \mathcal{A}_j(R_i, T_i^{-1}). \tag{19}$$

Then, we can obtain a new inequation by adding (18) and (19) together:

$$(\theta_i - \theta_j)\nu(R_i) \geq (\theta_i - \theta_j)\nu(R_j). \tag{20}$$

(a) Sufficiency Due to $\theta_i > \theta_j$, $\theta_i - \theta_j > 0$ is true. We can get $\nu(R_i) \geq \nu(R_j)$ by deriving from (20). As $\nu'(R_i) > 0$, we can conclude that $R_i > R_j$. So the sufficiency condition is proved.

(b) Necessity The inequation in (20) can be transformed and rewritten as

$$\theta_i(\nu(R_i) - \nu(R_j)) \geq \theta_j(\nu(R_i) - \nu(R_j)), \tag{21}$$

as $\nu'(R_i) > 0$ and $R_i > R_j$, we can conclude that $\theta_i > \theta_j$ easily. In addition, the $R_i = R_j$ and $\theta_i = \theta_j$ cases can be proved by the similar processes. ■

Proposition 1: $R_i \geq R_j$, if and only if $T_i^{-1} \geq T_j^{-1}$.

Subproof 1: According to the IC constraints expressed in (18) and (19), we rewrite them as

$$\begin{aligned}\theta_i(\nu(R_i) - \nu(R_j)) \geq \omega(T_i^{-1} - T_j^{-1}) \\ + \varepsilon(\rho(R_i) - \rho(R_j)),\end{aligned} \tag{22}$$

$$\begin{aligned}\omega(T_i^{-1} - T_j^{-1}) \geq \theta_j(\nu(R_i) - \nu(R_j)) \\ - \varepsilon(\rho(R_i) - \rho(R_j)).\end{aligned} \tag{23}$$

(a) Sufficiency Since $T_i^{-1} \geq T_j^{-1}$, a new inequation can be derived from (22), i.e.,

$$\theta_i(\nu(R_i) - \nu(R_j)) \geq \varepsilon(\rho(R_i) - \rho(R_j)), \tag{24}$$

where θ_i is monotonic increasing and ε is a fixed value that $\max\{\varepsilon\} \leq \theta_1$. So (24) can be transformed to

$$\big(\theta_1\nu(R_i) - \varepsilon\rho(R_i)\big) \geq \big(\theta_1\nu(R_j) - \varepsilon\rho(R_j)\big). \tag{25}$$

Set $f(R_i) = \theta_1\nu(R_i) - \varepsilon\rho(R_i)$, and then we have

$$f(R_i) \geq f(R_j), \tag{26}$$

where $f' = \theta_1\nu' - \varepsilon\rho'$. With the constraints $(c), (d)$ and $\nu' > \rho'$, we can conclude that $f' > 0$, and thus $R_i \geq R_j$.

(b) Necessity Since $R_i \geq R_j$, a new inequation can be derived from (23), i.e.,

$$\begin{aligned}\omega(T_i^{-1} - T_j^{-1}) \geq \big(\theta_j\nu(R_i) - \varepsilon\rho(R_i)\big) \\ - \big(\theta_j\nu(R_j) - \varepsilon\rho(R_j)\big).\end{aligned} \tag{27}$$

Obviously, this case can be proved by the similar steps mentioned in (22) and (23), so the subsequent proof is omitted here. ■

The Lemma 2 indicates that in the incentive compatibility contract, the validators with a lower voting time will gain a higher reward.

Lemma 3: If the IR constraint of type-1 validators is satisfied, the IR constraints of other types will hold as well.

Proof 3: The conditions listed in (15) indicate that all of IR constraints need to be satisfied. Nevertheless, according to the IC constraints (14), for $\forall i \in \{1, ..., N\}$, we first have

$$\begin{aligned}\theta_i(\nu(R_i)) - \omega T_i^{-1} - \varepsilon\rho(R_i) \geq \theta_i(\nu(R_1)) \\ - \omega T_1^{-1} - \varepsilon\rho(R_1).\end{aligned} \tag{28}$$

Then, on the basis of (15) and (28), we obtain a new inequation

$$\begin{aligned}\theta_i(\nu(R_1)) - \omega T_1^{-1} - \varepsilon\rho(R_1) \geq \theta_1(\nu(R_1)) \\ - \omega T_1^{-1} - \varepsilon\rho(R_1).\end{aligned} \tag{29}$$

Finally, given (28) and (29), we can conclude that

$$\begin{aligned}\theta_i(\nu(R_i)) - \omega T_i^{-1} - \varepsilon\rho(R_i) \geq \theta_1(\nu(R_1)) - \omega T_1^{-1} \\ - \varepsilon\rho(R_1) \geq 0.\end{aligned} \tag{30}$$

■

With the IC constraint, the Lemma 3 indicates that only when the IR constraint of type-1 is kept, can the others be also satisfied.

Lemma 4: Here are four definitions regarding the IC constraints between type-i and type-j:

(a) If $\forall j \in \{1, ..., i-1\}$, the constraints are called Downward Incentive Constraints (**DIC**s).
(b) If $j = i - 1$, the constraint is called Local Downward Incentive Constraint (**LDIC**).
(c) If $\forall j \in \{i+1, ..., N\}$, the constraints are called Upward Incentive Constraints (**UIC**s).
(d) If $j = i + 1$, the constraint is called Local Upward Incentive Constraint (**LUIC**).

With the monotonicity proved in Lemma 2, the DICs can be reduced as LDICs and the UICs can be reduced as the LUICs.

Proof 4: All of the validators are classified into different types, and there exists the IC constraint between any two types. As a result, there are too many IC constraints in total, which will increase the difficulty of computation. Here we will prove that all of the IC constraints can be reduced as LDICs. Consider three adjacent types, i.e., type $i-1$, type i and type $i+1$, which follows $\forall i \in \{1, ..., N-1\}$. According to the IC constraints, then we have the following two inequations:

$$\begin{aligned}\theta_{i+1}(\nu(R_{i+1})) - \omega T_{i+1}^{-1} - \varepsilon\rho(R_{i+1}) \geq \theta_{i+1}(\nu(R_i)) \\ - \omega T_i^{-1} - \varepsilon\rho(R_i),\end{aligned} \tag{31}$$

$$\begin{aligned}\theta_i(\nu(R_i)) - \omega T_i^{-1} - \varepsilon\rho(R_i) \geq \theta_i(\nu(R_{i-1})) \\ - \omega T_{i-1}^{-1} - \varepsilon\rho(R_{i-1}).\end{aligned} \tag{32}$$

Since the monotonicity has been proved in lemma 1, i.e., $R_i \geq R_j$ and $T_i^{-1} \geq T_j^{-1}$ if and only if $\theta_i \geq \theta_j$. Set $\mathcal{A}_x(R_y, T_y^{-1}) = \theta_x\nu(R_y) - \omega T_y^{-1} - \varepsilon\rho(R_y)$. We derive a new inequation from (32):

$$\mathcal{A}_{i+1}(R_i, T_i^{-1}) \geq \mathcal{A}_{i+1}(R_{i-1}, T_{i-1}^{-1}). \tag{33}$$

According to the inequations (31) and (33), it can be easily obtained:

$$\mathcal{A}_{i+1}(R_{i+1}, T_{i+1}^{-1}) \geq \mathcal{A}_{i+1}(R_{i-1}, T_{i-1}^{-1}). \tag{34}$$

Repeat the procedures listed above for other types, we can get the following constraints from (34):

$$\begin{aligned}\mathcal{A}_{i+1}(R_{i+1},T_{i+1}^{-1}) &\geq \mathcal{A}_{i+1}(R_{i-1},T_{i-1}^{-1})\\ &\geq ...\\ &\geq \mathcal{A}_{i+1}(R_1,T_1^{-1})\\ &\geq \mathcal{A}_1(R_1,T_1^{-1}).\end{aligned} \tag{35}$$

Therefore, the proof indicates that if the LDIC is satisfied, all the DICs can also hold.

Similarly, according to the IC constraints, we have

$$\mathcal{A}_{i-1}(R_{i-1},T_{i-1}^{-1}) \geq \mathcal{A}_{i-1}(R_i,T_i^{-1}), \tag{36}$$

$$\mathcal{A}_i(R_i,T_i^{-1}) \geq \mathcal{A}_i(R_{i+1},T_{i+1}^{-1}). \tag{37}$$

The same procedures are omitted here. Finally we can easily come to:

$$\begin{aligned}\mathcal{A}_{i-1}(R_{i-1},T_{i-1}^{-1}) &\geq \mathcal{A}_{i-1}(R_{i+1},T_{i+1}^{-1})\\ &\geq ...\\ &\geq \mathcal{A}_{i-1}(R_N,T_N^{-1}).\end{aligned} \tag{38}$$

Hence, we complete the proof that if the LUIC is satisfied, all the UICs hold as well. ■

With the reduced constraints, we can redefine the optimization problem as follows:

$$\begin{aligned}\max_{(R_i,T_i^{-1})} \quad U_{BC} = \sum_{i=1}^{\mathbb{N}} \lambda_i \bigg(a_1 \Big(\frac{\theta_i}{\bar{\theta}}\Big)^{\alpha} + a_2(\sum_{j=1}^{i}\theta_j^{-2}-d)\\ \Big(\frac{\rho(R_i)}{\rho(\bar{R})}\Big)^{\beta} - a_3\Big(\frac{1}{T_{max}T_i^{-1}}\Big)^{\gamma} - \mu R_i\bigg),\end{aligned} \tag{39}$$

s.t.

$$\begin{aligned}&(a) \quad \mathcal{A}_1(R_1,T_1^{-1}) = 0,\\ &(b) \quad \mathcal{A}_i(R_i,T_i^{-1}) = \mathcal{A}_i(R_{i-1},T_{i-1}^{-1}), \forall i \in \{2,...,N\},\\ &(c) \quad 0 < \theta_1 < ... < \theta_i < ... < \theta_N,\\ &(d) \quad \max\{T_i\} \leq \min\{t_{max},t_0\}.\end{aligned}$$

To solve this problem, according to the constraints (a) and (b) in (III-B), $\forall i \in \{2,...,N\}$, we first set

$$\Delta_i = \theta_i\bigg(\nu(R_i)-\nu(R_{i-1})\bigg) - \varepsilon\bigg(\rho(R_i)-\rho(R_{i-1})\bigg), \tag{40}$$

and then we derive a new equation from (a) and (b) in (III-B):

$$T_i^{-1} = \frac{\theta_1\nu(R_1)-\varepsilon\rho(R_1)+\sum_{j=2}^{i}\Delta_j}{\omega}. \tag{41}$$

Before substituting T_i^{-1} into (III-B), we transform equation (41), as follows

$$T_i^{-1} = \frac{\theta_i\nu(R_i)-\varepsilon\rho(R_i)}{\omega} + \kappa_{i-1}, \tag{42}$$

where

$$\kappa_{i-1} = \begin{cases} \frac{\sum_{j=1}^{i-1}\nu(R_j)(\theta_j-\theta_{j+1})}{\omega}, & 2 \leq i \leq \mathbb{N},\\ 0, & i=1.\end{cases} \tag{43}$$

After substituting (43) into (III-B), (III-B) is converted into a new problem:

$$\begin{aligned}\max_{(R_i,T_i^{-1})} \quad U_{BC} = \sum_{i=1}^{\mathbb{N}} \lambda_i \bigg(a_1 \Big(\frac{\theta_i}{\bar{\theta}}\Big)^{\alpha} + a_2(\sum_{j=1}^{i}\theta_j^{-2}-d)\\ \Big(\frac{\rho(R_i)}{\rho(\bar{R})}\Big)^{\beta} - a_3\Big(\frac{1}{T_{max}T_i^{-1}}\Big)^{\gamma} - \mu R_i\bigg),\end{aligned} \tag{44}$$

s.t.

$$\begin{aligned}&(a) 0 < \theta_1 < ... < \theta_i < ... < \theta_N,\\ &(b) \max\{T_i\} \leq \min\{t_{max},t_0\}.\end{aligned}$$

We differentiate U_{BC} with respect to R_i and then get

$$\begin{aligned}\frac{\partial U_{BC}(i)}{\partial R_i} = \lambda_i\bigg\{\frac{a_2\beta(\rho(R_i))^{\beta}(\sum_{j=1}^{i}\theta_j^{-2}-d)}{\rho(\bar{R})^{\beta}}\frac{\partial\rho(R_i)}{\partial R_i}\\ + \frac{a_3\gamma}{T_{max}T_i^{-2}}\frac{\partial(T_i^{-1})}{\partial R_i} - \mu\bigg\}.\end{aligned} \tag{45}$$

Next, by differentiating $\frac{\partial U_{BC}}{\partial R_i}$ with respect to R_i, we have

$$\begin{aligned}\frac{\partial^2 U_{BC}(i)}{\partial (R_i)^2} =&\lambda_i\bigg\{\frac{a_2\beta(\beta-1)(\sum_{j=1}^{i}\theta_j^{-2}-d)}{(\rho(R_i))^{2-\beta}\rho(\bar{R})^{\beta}}(\frac{\partial\rho(R_i)}{\partial R_i})^2\\ &+ \frac{a_2\beta(\rho(R_i))^{\beta-1}}{(\sum_{j=1}^{i}\theta_j^{-2}-d)\rho(\bar{R})^{\beta}}\frac{\partial^2\rho(R_i)}{\partial(R_i)^2}\\ &- \frac{2a_3\gamma}{T_{max}T_i^{-3}}(\frac{\partial(T_i^{-1})}{\partial R_i})^2\\ &+ \frac{a_3\gamma}{T_{max}T_i^{-2}}\frac{\partial^2(T_i^{-1})}{\partial(R_i)^2}\bigg\}.\end{aligned} \tag{46}$$

Since $\beta < 1, 0 \leq \rho' < \nu'$ and $\nu'' \leq \rho'' \leq 0$, we have $\frac{\partial T_i^{-1}}{\partial R_i} > 0$ and $\frac{\partial^2 T_i^{-1}}{\partial(R_i)^2} < 0$, and thus we can get

$$\frac{\partial^2 U_{BC}}{\partial(R_i)^2} < 0. \tag{47}$$

Obviously, (47) denotes that the optimal problem function is a concave function. Our goal is to figure out the maximum value of a concave function with the affine constraints, which is a convex optimization problem. From (47) we can conclude that, there must exit the values R_i^* and T_i^{-1*} that make the following equation hold.

$$\begin{aligned}\frac{\partial U_{BC}}{\partial R_i} = \lambda_i\bigg\{\frac{a_2\beta(\rho(R_i^*))^{\beta}(\sum_{j=1}^{i}\theta_j^{-2}-d)}{\rho(\bar{R})^{\beta}}\frac{\partial\rho(R_i^*)}{\partial R_i}\\ + \frac{a_3\gamma}{T_{max}T_i^{-2*}}\frac{\partial(T_i^{-1*})}{\partial R_i} - \mu\bigg\} = 0.\end{aligned} \tag{48}$$

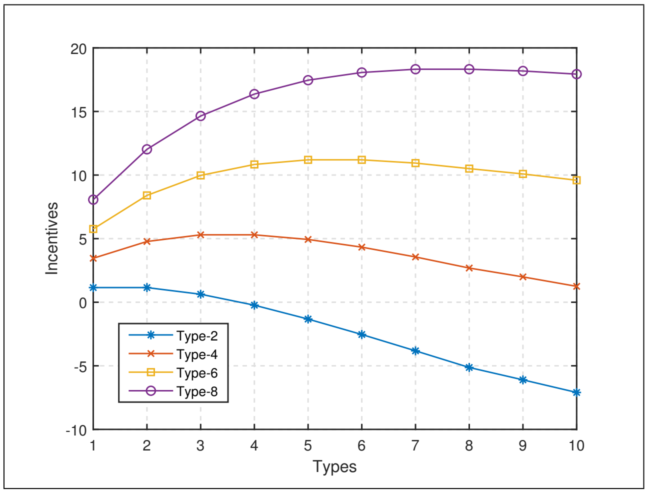

Fig. 2. Utilities of validators when sign different contracts

TABLE I
PARAMETERS

Parameter	Value
Weight Parameters	$a_1 = 20, a_2 = 50, a_3 = 300$
Exponent Parameters	$\alpha = 2, \beta = 0.5, \gamma = 1, d = 0.5$
Default type and reward	$\bar{\theta} = 5, \bar{R} = 32/5$
t_{next}	$300s$
Unit Cost	$\mu = 2, \omega = 0.01$
Probability Parameter	$\lambda_i = 0.1$
Total Types	$types \in \{1, ..., 10\}$

Therefore, we can obtain the optimal reward R_i^* and the related time T_i^{-1*} by convex optimization tools. Moreover, $\rho(R_i^*)$ for the type-i validators can be determined as well, which is set as their security deposit.

IV. Numerical Results and Analysis

In this section, we first evaluate the performance of the proposed contract-based scheme, proving the feasibility in such a scenario. Next, we compare and discuss the possible deposit value with the fixed value 32ETH. In the parameters setting part, we assume that there are 10 types in total, the other related parameters are listed in Table 1. For further analysis without loss the generality, we set $\nu(R_i) = R_i$ and $\rho(R_i) = 2R_i$. Besides, we set a convert function $\varphi(R_i) = \mathcal{C}_{ETH}$ that converts the rewards into ETH, here we set $\varphi(R_i) = 5R_i$. Then we present the numerical results in the following steps.

First, the beacon chain acts as the contract designer and define the different types and the correlative rewards and time. Then, all of the validators receive the contracts and choose one of them according to their own types. Finally, the beacon chain will distribute the payoffs to the validators who complete their work as required.

As shown in Fig. 2, the validators of type-2, type-4, type-6 and type-8 signing the different contracts have various utilities.

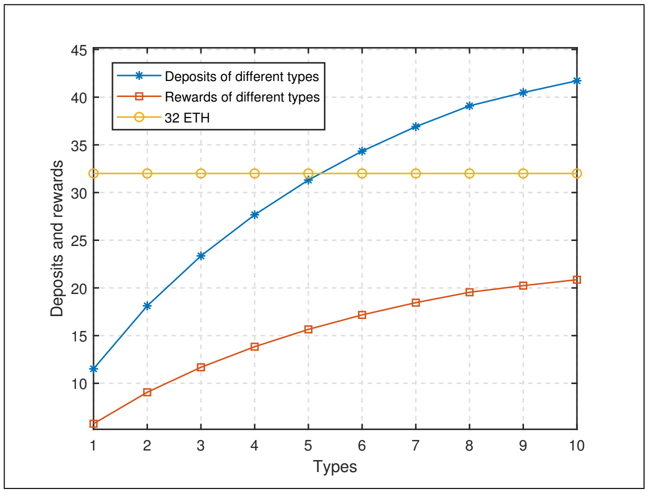

Fig. 3. The comparison among the deposits, rewards and 32 ETH.

It shows apparently that the validators have the maximum utilities only when choosing the contract designed for their own, which prove the IC constraint. Besides, all these maximum incentives are positive, which explains the IR constraint.

We also compare and discuss the rewards, deposits and the fixed value set by Ethereum in the following simulation. By utilizing the convert function, we set the 32ETH as one of the parameters in the problem, and then get the optimal result shown in Fig. 3. We can figure out that the validators of higher type have more rewards, also means the more deposits. For the validators of lower types, they do not need to submit 32ETH at all. According to their lower rewards, they are only required to submit a lower deposit. Since they gain so little in the network, and the high deposits lack economic incentives for such validators, so the lower deposits related to their rewards are acceptable. For the validators of higher types, they should hand in a number of deposits higher than 32 ETH. Since they can get more payoff than other lower types, the fixed deposit 32 ETH cannot provide enough security incentive for the network. As a result, we can come to the conclusion that the proposed scheme balances the security incentive and economic incentive for both sides.

V. Conclusion

In this paper, we propose a contract-based approach to balance the security incentive and economic incentive for the network and the validators. Specially, we formulate the problem under the contract theory framework and classify the validators into different types according to their performance and stakes. It can increase the economic incentive for the validators with less performance and stakes, and increase the security incentive for the network by increasing the deposit of the validators with high performance and stakes. The proposed scheme can overcome the information asymmetry introduced by the inherent features of the blockchain. As a result, this approach allows the validators to obtain their rewards and submit the deposits according to their types. Moreover, even though the validators of higher types have to submit the higher deposits, they have more rewards to motivate them to do so. The simulation results exactly show that the optimal contract-based approach can achieve the balance between the security incentive and economic incentive.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Grant Nos. 61702258, in part by US MURI, US NSF CNS-1717454, CNS-1731424, CNS-1702850 and CNS-1646607.

References

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Self-published Paper*, 2008 [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[2] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services*, 14(4):352–375, 2018.

[3] Wenbo Wang, Dinh Thai Hoang, Zehui Xiong, Dusit Niyato, Ping Wang, Peizhao Hu, and Yonggang Wen. A survey on consensus mechanisms and mining management in blockchain networks. *arXiv preprint arXiv:1805.02707*, 2018.

[4] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151:1–32, 2014.

[5] QuantumMechanic. Proof of stake instead of proof of work. https://bitcointalk.org/index.php?topic=27787.0. June 11, 2011.

[6] Iddo Bentov, Ariel Gabizon, and Alex Mizrahi. Cryptocurrencies without proof of work. In *International Conference on Financial Cryptography and Data Security*, pages 142–157. Springer, 2016.

[7] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*, pages 357–388. Springer, 2017.

[8] James Ray. Sharding introduction rd compendium. https://github.com/ethereum/wiki/wiki/Sharding-introduction-R&D-compendium. Oct 17, 2018.

[9] Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, et al. On scaling decentralized blockchains. In *International Conference on Financial Cryptography and Data Security*, pages 106–125. Springer, 2016.

[10] Loi Luu, Viswesh Narayanan, Chaodong Zheng, Kunal Baweja, Seth Gilbert, and Prateek Saxena. A secure sharding protocol for open blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 17–30. ACM, 2016.

[11] Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, and Bryan Ford. Omniledger: A secure, scale-out, decentralized ledger. *IACR Cryptology ePrint Archive*, 2017:406, 2017.

[12] Vitalik Buterin and Virgil Griffith. Casper the friendly finality gadget. *arXiv preprint arXiv:1710.09437*, 2017.

[13] Vlad Zamfir. Introducing casper the friendly ghost. https://blog.ethereum.org/2015/08/01/introducing-casper-friendly-ghost/. Aug 1, 2015.

[14] Ethereum 2.0 phase 0 – the beacon chain. https://github.com/ethereum/eth2.0-specs/blob/master/specs/core/0_beacon-chain.md#deposit-contract.

[15] Tingting Liu, Jun Li, Feng Shu, Yongpeng Wu, Zhu Han, et al. Incentive mechanism design for two-layer wireless edge caching networks using contract theory. *IEEE Transactions on Services Computing*, 2018.

[16] Tingting Liu, Jun Li, Feng Shu, Meixia Tao, Wen Chen, and Zhu Han. Design of contract-based trading mechanism for a small-cell caching system. *IEEE Transactions on Wireless Communications*, 16(10):6602–6617, 2017.

[17] Patrick Bolton, Mathias Dewatripont, et al. *Contract theory*. MIT press, 2005.

[18] Xiaojun Liu, Wenbo Wang, Dusit Niyato, Narisa Zhao, and Ping Wang. Evolutionary game for mining pool selection in blockchain networks. *IEEE Wireless Communications Letters*, 2018.