SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

October 2021, Vol. 64 202301:1–202301:21
https://doi.org/10.1007/s11432-019-2858-8

An incentive-compatible rational secret sharing scheme using blockchain and smart contract

Zerui CHEN[1], Youliang TIAN[1,2*] & Changgen PENG[1,2]

[1] *State Key Laboratory of Public Big Date, College of Computer Science and Technology, Guizhou University, Guiyang* 550025*, China;*
[2] *Institute of Cryptography and Data Security, Guizhou University, Guiyang* 550025*, China*

Received 2 December 2019/Revised 14 March 2020/Accepted 18 March 2020/Published online 3 February 2021

Abstract In the rational cryptographic protocol, the two rational players often fall into the prisoner's dilemma, which is also the case for the rational secret sharing we consider in this paper. First, it is proved that rational secret sharing has a sequential equilibrium in the natural state, so that rational participants will fall into the prisoner's dilemma, resulting in no participants being able to reconstruct the secret correctly. Next, to solve this problem, we propose an incentive-compatible rational secret scheme. Specifically, the game tree with imperfect information is constructed to facilitate our analysis and proof, and the strictly dominated strategies are directly eliminated to simplify the game tree. Further more, we describe the motivation of the verifier. Then, we prove that rational players have no motivation to deviate from honest behavior using sequential equilibrium so that rational players can reconstruct the secret correctly. Finally, we complete the simulation using the smart contract and analyze our entire scheme. In addition, the game of our scheme does not need to be repeated multiple times to reach sequential equilibrium, i.e., the game always follows the rational path.

Keywords rational secret sharing, game theory, sequential equilibrium, incentive-compatible, smart contract

Citation Chen Z R, Tian Y L, Peng C G. An incentive-compatible rational secret sharing scheme using blockchain and smart contract. Sci China Inf Sci, 2021, 64(10): 202301, https://doi.org/10.1007/s11432-019-2858-8

1 Introduction

In 1979, Blakey [1] and Shamir [2] separately studied the well-known t-out-of-n secret sharing scheme. Their basic idea is that a dealer who holds a secret divides the secret into n sub-secrets and distributes the secret shares to n participants, and any t or more participants can reconstruct the secret, while less than t players cannot. In traditional secret sharing, we think participants are either "good" or "bad", and "good" participants are always willing to participate in reconstruction honestly, while "bad" participants are unwilling to do so. Then Halpern and Teague [3] used the game theory to analyze secret sharing and put forward rational secret sharing scheme for the first time. Dodis and Rabin [4] pointed out that the game theory and cryptography can be effectively combined if a reasonable scheme can be designed. In such circumstances, participants are neither "good" nor "bad". To be precise, they are rational, i.e., their purpose is to maximize their own utility.

1.1 Related work

Halpern and Teague [3] studied the Nash equilibrium that was determined by iterated deletion of weakly-dominated strategies in rational secret sharing. Then Gordon and Katz [5] extended the scheme [3] and gave a Nash equilibrium solution with the number of participants $n = 2$ in rational secret sharing, however, not all bad strategies can be eliminated. Kol and Naor [6] proposed the strict Nash equilibrium, but the equilibrium concepts were too difficult to achieve. Fuchsbauer et al. [7] presented computational

[*] Corresponding author (email: youliangtian@163.com)

Nash equilibrium stable with respect to trembles, and they allowed the mistakes of the parties. Like the idea of solving the "repeated prisoner's dilemma" in [8], some scholars repeated a secret sharing multiple times, and guaranteed the participants' honest behavior through an incentive mechanism. For example, Maleka et al. [9] studied the repeated games model of rational secret sharing scheme. However, repeated game requires the game to be played multiple times, and if the participants know that they are in the last sub-game of secret reconstruction, they have no motivation to be honest at this time. Ong et al. [10] studied the subgame perfect equilibrium, but a small number of honest players should be assumed in their model.

Besides, Zhang and Liu [11] presented information-theoretic secure rational secret sharing scheme in a standard communication network. Then, they designed a credible punishment mechanism [12] in rational secret sharing based on the extensive game. By introducing rational communication players, Tian et al. [13] proposed a formal framework to solve the problem of interaction among distrusted players. Then, they studied the utility function of rational players, and presented a new rational secret sharing scheme [14] based on Bayesian game. Jin et al. [15] proposed a rational secret sharing scheme based on the reputation mechanism, and their protocol only required one round. By introducing mechanism design [16] from the field of microeconomics, Liu et al. [17] proposed a reasonable secret reconstruction protocol in rational secret sharing.

Nakamoto [18] first proposed the data structure of blockchain and a virtual crypto-currency Bitcoin in 2008, which guaranteed neutralization, undeniableness, uniqueness and traceability. Recently, some scholars have done work on the combination of blockchain and secret sharing (see [19–22]), and the focus of their study is on implementing some mechanics of blockchain by using traditional secret sharing. For example, Bartolucci et al. [20] used secret sharing to enable on-chain and introduced a secret share-based voting system on the blockchain. However, our study is to achieve an incentive-compatible rational secret sharing scheme using blockchain and smart contract. In addition, from the study of Dong et al. [23], it is feasible to realize the incentive mechanism in rational cryptography through smart contracts.

1.2 Our contribution

In fact, we also have the question, once game theory is introduced, how will the traditional secret sharing protocol analysis be affected? We consider the players in the secret sharing are rational and define a utility function for each player whose purpose is to maximize his/her own utility. Unfortunately, as pointed out in [3], no rational player has an incentive to honestly publish his/her secret share during a secret reconstruction, which results in no player being able to reconstruct the secret, this phenomenon is called the prisoner's dilemma in game theory. And as pointed out in [24], certain game-theoretic equilibria are achievable if a trusted mediator is available, so we introduce an additional trusted verifier. Until now, a lot of the cryptographic work has been evaluated in terms of game theory. However, many existing rational secret sharing schemes contain some incredible threats, leading to the Nash equilibrium is actually unreasonable. Our contributions are mainly as follows.

(1) We propose an incentive-compatible rational secret sharing scheme. Specifically, we redesigned the process of secret sharing, i.e., the players not only need to send the secret share, but also need to send the reconstructed secret after reconstructing the secret for verifiability. Note that the players only need to send the commitment of secret to ensure privacy. Then we design the incentive mechanism and guarantee the rational players' honesty in each phase. In addition, we rigorously prove that there is a unique sequential equilibrium in the game and the game always follows the rational path. More precisely, we define players' utility for each step, and then we prove that players have no motivation to deviate from honest behavior in order to maximize their own utility.

(2) Our research is based on an extensive game rather than a strategic game, which helps us to provide credible punishment strategies. Moreover, we consider the imperfect information game, which is close to the reality. Then we eliminate strictly dominated strategies to simplify the game tree that can facilitate our analysis and proof.

(3) We complete the simulation using the smart contract, which provids the possibility for the application of our scheme in realistic secret sharing scenarios. In addition, we analyze our entire scheme and conclude that the overhead of the smart contract we design and deploy is extremely small.

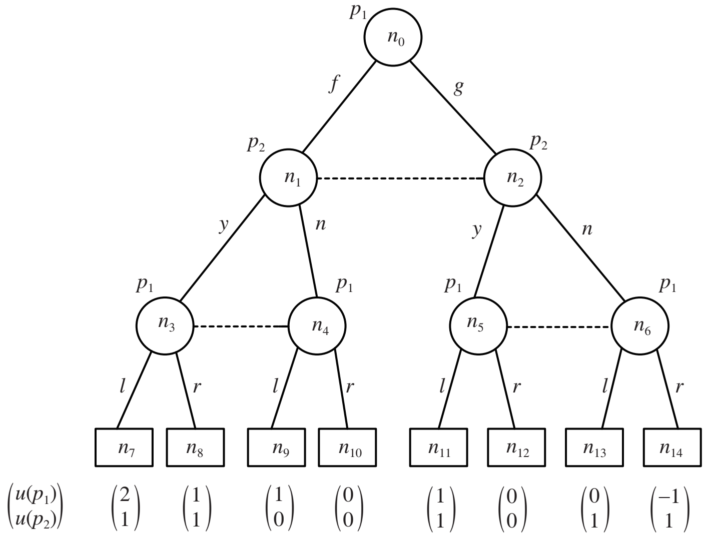

Figure 1 GameSample.

2 Preliminaries

In this section, we introduce the extensive game with imperfect information. We review the Nash equilibrium and introduce a improvement that has strong influence in equilibrium, which we call sequential equilibrium.

2.1 Game theory

Our paper is based on the extensive game framework. Note that all the games in our paper are finite. Often in a game, the actions of the players are in order, the players only know the action strategies of other players, but do not know what actions other players specifically choose. So the extensive game with imperfect information is closer to reality and has broader practical implications, which is why we consider it. We construct the game tree to describe the subsequent game. Through the game tree, we can clearly see the players, nodes, information sets, optional actions, utility functions, etc. of a game.

Definition 1. An extensive game with imperfect information is a $G = \{P, A, H, E, I, Q, U\}$, where

- P is a set of players;
- A is a set of players' actions;
- H is a set of non-terminal selection nodes;
- E is a set of terminal nodes;
- I is information set;
- Q is a set of optional actions of players in the information set;
- U is a set of utility functions.

As shown in Figure 1, we use circles to represent the elements in H, that is, the non-terminal selection nodes. We use rectangles to represent the elements in E, i.e., the terminal nodes. We use n_i to represent the node number. We also use horizontal lines to represent the nodes reached by the players after the action is selected, and use dotted lines to connect non-terminal selection nodes at the same information set. When the terminal node is reached, the player's utility function is displayed as follows.

- $P = \{p_1, p_2\}$ represents the players of the game are p_1, p_2.
- $A = \{f, g, y, n, l, r\}$ represents the combination of all the action strategies of all players.
- $H = \{n_0, n_1, n_2, n_3, n_4, n_5, n_6\}$ represents that when the game reaches node $n_0 - n_6$, the game is still going on, and the player at the corresponding node must make a choice of action.
- $E = \{n_7, n_8, n_9, n_{10}, n_{11}, n_{12}, n_{13}, n_{14}\}$ represents that when the game reaches node $n_7 - n_{14}$, the game ends.
- $I = \{I_{11}, I_{12}, I_{13}, I_{21}\}$. The information set of player p_1 is I_{11}, I_{12}, I_{13} where $I_{11} = \{n_0\}, I_{12} = \{n_3, n_4\}, I_{13} = \{n_5, n_6\}$, and the information set of player p_2 is I_{21} where $I_{21} = \{n_1, n_2\}$. For example, when the player p_2 is at information set I_{21}, he/she cannot know whether he/she is at n_1 or n_2. In other words, at this time, he/she only knows that player p_1 has made an action at the starting node n_0,

however, he/she cannot know exactly whether p_1 has chosen the action f or the action g, which is the embodiment of imperfect information.

- $Q = \{Q_{11}, Q_{12}, Q_{13}, Q_{21}\}, Q_{11} = \{f, g\}$,$Q_{12} = Q_{13} = \{l, r\}$,$Q_{21} = \{y, n\}$ where Q_{11} is an optional action when player p_1 is at information set I_{11}, Q_{21} is an optional action when p_2 is at I_{21}, Q_{12} and Q_{13} are optional actions when p_1 is at I_{12} or I_{13}.
- $U = \{u_7, u_8, u_9, u_{10}, u_{11}, u_{12}, u_{13}, u_{14}\}$ represents when the game reaches the terminal node $n_7 - n_{14}$, the corresponding utility function is $u_7 - u_{14}$, such as $u_7 = (2, 1)$ represents when the game reaches the terminal node n_7, p_1 gets utility 2 and p_2 gets utility 1. Note that the purpose of rational players is to maximize their own utility.

Definition 2. In a game G, an optional strategy combination s_i of player p_i means that for $\forall I_{ij} \in I_i$, we assign probabilities to all optional actions $Q_{ij} \in Q_i$.

For example, as shown in Figure 1, $s_1 = ([\frac{1}{2}(f), \frac{1}{2}(g)], [\frac{1}{3}(l), \frac{2}{3}(r)], [\frac{2}{3}(l), \frac{1}{3}(r)])$ is an optional strategy combination of players p_1, which means that when player p_1 arrives at information set I_{11}, he/she selects action f with probability $\frac{1}{2}$, selects action g with probability $\frac{1}{2}$; when he/she arrives at information set I_{12}, he/she selects action l with probability $\frac{1}{3}$, selects action r with probability $\frac{2}{3}$; when he/she arrives at information set I_{12}, he/she selects action l with probability $\frac{2}{3}$, selects action r with probability $\frac{1}{3}$.

Definition 3. In a game G, the strategy combination S_i of player p_i is a set of all optional strategy combinations of player p_i:

$$S_i = (s_{i1}, s_{i2}, \ldots, s_{in}).$$

Definition 4. In a game G, the strategy profile S of the game means that the Cartesian product is made on the strategy combinations $(S_1, S_2, \ldots, S_n)$ of all players:

$$S = S_1 \times S_2 \times \cdots \times S_n,$$

where n means the number of players.

Definition 5. In a game G, an optional strategy profile $s \in S$ of the game means assigning a probability to all players' optional actions at all information sets:

$$s = (s_i, s_{-i}),$$

where $s_{-i} = (s_1, s_2, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$ and n means the number of players.

For example, as shown in Figure 1, $s = (s_1, s_2)$ is an optional strategy profile of the game. Where $s_1 = ([\frac{1}{2}(f), \frac{1}{2}(g)], [\frac{1}{3}(l), \frac{2}{3}(r)], [\frac{2}{3}(l), \frac{1}{3}(r)]), s_2 = ([0(y), 1(n)])$ mean for player p_1, when information set I_{11} is reached, action f is selected with probability $\frac{1}{2}$ and action g is selected with probability $\frac{1}{2}$; when information set I_{12} is reached, action l is selected with probability $\frac{1}{3}$ and action r is selected with probability $\frac{2}{3}$; when information set I_{13} is reached, action l is selected with probability $\frac{2}{3}$ and action r is selected with probability $\frac{1}{3}$; and when player p_2 arrives at information set I_{21}, he/she must choose action n and never choose action y.

2.2 Nash equilibrium and sequential equilibrium

The most important concept in game theory is the Nash equilibrium. However, Nash equilibrium is not refined in many cases and some Nash equilibrium in the game may actually have incredible threats, and thus the Nash equilibrium is unreasonable. In order to eliminate these incredible threats and to refine the Nash equilibrium, some improvements to the Nash equilibrium are needed. Sequential equilibrium is considered to be a powerful improvement, even with a stricter definition than perfect Bayesian equilibrium.

Definition 6. In a game G, a Nash equilibrium $s^* = (s_i^*, s_{-i}^*)$, $s^* \in S$ refers to the fact that when one player deviates from the Nash equilibrium, it is impossible for him/her to obtain higher utility.

$$\forall s_i \in S_i, \quad u_i(s_i, s_{-i}^*) \leqslant u_i(s_i^*, s_{-i}^*).$$

In other words, when the strategy profile of the game reaches the Nash equilibrium, the strategy combination of each player is the best response for the given strategy combination of the other players, and no one can get higher utility by changing his/her own strategy combination.

Sequential equilibrium eliminates bad strategies by requiring that the game is sequential rationality, that is, it not only achieves optimal results in the final result of the game, but also achieves optimal results at each information set. In an extensive game with imperfect information, sequential equilibrium consists of a strategy profile and belief system. The strategy profile has been described in the previous article and will not be described again.

The belief system refers to the judgment of player p_i on a specific non-terminal selection node at the information set in which it is located. In the game tree, when p_i is at a certain information set, he/she judges the probability of being at the specific node. The player needs a belief system to determine his/her strategy combination so that it is optimal at each information set.

Definition 7. In a game G, belief system β_i of player p_i means that for $\forall I_{ij} \in I_i$, player p_i holds belief $\beta_i(x) = \mathrm{pr}[x|I_{ij}], x \in I_{ij}$.

The belief system indicates that when a player is at an arbitrary information set, he/she is convinced of the probabilities distribution of the specific nodes at the information set.

Definition 8. In a game G, when the strategy profile is s, the expected utility of the player p_i at node x is, for the reachable terminal at this time, the sum of the utility at each terminal node:

$$u_i(s, x) = \sum_{e \in E} u_i(e) \cdot \mathrm{pr}[e|s, x],$$

where $u_i(e)$ represents the p_i's utility at the terminal node and $\mathrm{pr}[e|s, x]$ represents the probability of arriving at the reachable terminal node $n_i \in E$ from the node x when the strategy profile is s.

Definition 9. In a game G, when the belief system of player p_i is β_i, the expected utility of player p_i at I_{ij} is the sum of expected utility at each $x \in I_{ij}$.

$$u_i(s, \beta, I_{ij}) = \sum_{x \in I_{ij}} \beta_i(x) \cdot u_i(s, x).$$

Definition 10. In a game G, when β is a belief system, the strategy profile $s = (s_i, s_{-i})$ is called rational at the information set when

$$\forall s_i' \in S_i, \quad u_i((s_i', s_{-i}), \beta, I_{ij}) \leqslant u_i(s, \beta, I_{ij}).$$

Definition 11. In a game G, a (s, β) is called sequential rationality if for $\forall p_i \in P, I_{ij} \in I$, the strategy profile $s = (s_i, s_{-i})$ is rational at the information set.

Definition 12. In a game G, a (s, β) is called sequential consistency if there exists a sequence of fully mixed behavior strategy profile $(s^k)_{k \in N}$ converging to s and the sequence of belief $(\beta^k)_{k \in N}$ induced by $(s^k)_{k \in N}$ converging to the belief system β. In other words, it satisfies both

$$(1) \lim_{k \to \infty} (s^k) \to s, \quad (2) \lim_{k \to \infty} (\beta^k) \to \beta.$$

Definition 13. In a game G, a (s, β) is called the sequential equilibrium if it satisfies sequential rationality and consistency.

For a given sequential equilibrium (s, β), the player's expected utility at any point is the highest, and any strategy combination of the players that deviates from the sequential equilibrium does not make the utility higher. Therefore, rational players have no motivation to deviate from sequential equilibrium.

Definition 14. In a game G, $\mathrm{RPath}(x)$ is called the rational path if there is only one sequential equilibrium so that the game will always follow this unique path and end at the terminal node x.

3 Natural state and prisoner's dilemma

In this section, we introduce the rational secret sharing in the natural state. We prove that rational players are bound to fall into the prisoner's dilemma, so that no player has a motivation to honestly publish his/her secret share, which results in no player being able to reconstruct the secret.

3.1 Natural system model

In the nature state, a rational secret sharing process contains a secret generation center (SGC) and players who are called secret sharing parties (SSPs).

- SGC: The task of trusted SGC is to distribute and save the shared secret in the system, which is similar to a key generation center, and once the secret is distributed, it goes offline.

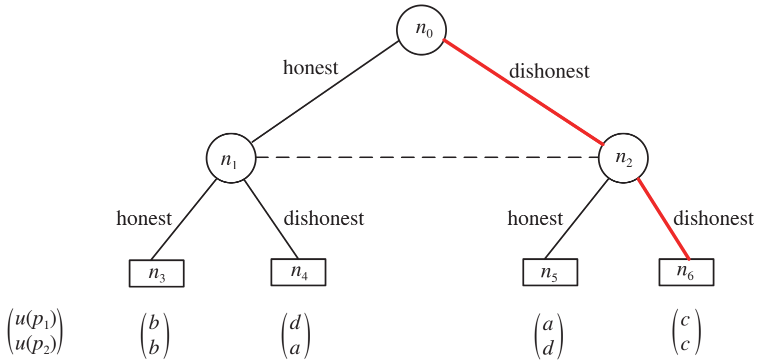

Figure 2 Game1.

- SSPs: The SSPs in our paper are rational p_1, p_2 in secret sharing and they can interact with each other.

And we describe such a process as the following two phases.

Distribution phase.

(1) The SGC randomly generates the shared key $W \in \{0,1\}^k$, and then it randomly selects w_1,$w_2 \in \{0,1\}^k$ and satisfies $W = w_1 \oplus w_2$.

(2) The SGC sends w_1 to p_1 and w_2 to p_2, and then it goes offline.

w_1 and w_2 are secret shares, obviously, if a player does not have two secret shares at the same time and k is large enough, and then the probability of getting W is an negligible $\xi = (\frac{1}{2})^k$.

Reconstruction phase.

(1) p_1 sends $w_1'/\bot$ to p_2 and p_2 sends $w_2'/\bot$ to p_1.

(2) p_1 computes $W_1 = w_1 \oplus w_2'$ and p_2 computes $W_2 = w_2 \oplus w_1'$.

For example, w_1' means p_1 sends a message to p_2, but this message may not be correct. $\bot$ means player does not send any message.

3.2 Value variables

To describe the behavior of rational players in the natural state, we define some value variables as follows.

- a : The utility of the player who reconstructs the secret correctly (if only one player reconstructs the secret correctly).
- b : The utility of each player (if both players reconstruct the secret correctly).
- c : The utility of each player (if no player reconstructs the secret correctly).
- d : The utility of the player who does not reconstruct the secret correctly (if only one player reconstructs the secret correctly).

Obviously, $a > b > c > d$.

3.3 Problem statement

Rational p_1, p_2 aim to maximize their own utility, during the reconstruction phase, they have three strategies: {publish}, {silent}, {cheat}.

{publish} means the player publishes his/her secret share honestly.

{silent} means the player does not publish his/her secret share.

{cheat} means the player publishes his/her secret share dishonestly.

Because {silent} and {cheat} have no effect on utility functions, and the both strategies represent dishonest behaviors, so they are called {dishonest}. Similarly, {publish} is called {honest}.

When p_i sends the real $w_i' = w_i$ to the other player, if he/she receives the wrong $w_{-i}' \neq w_{-i}$ or even does not receive a secret share, then he/she gets the worst utility d; if he/she receives the right $w_{-i}' = w_{-i}$, then he/she gets utility b which is less than utility a. In fact, at this time, the strategy {honest} is strictly inferior to the strategy {dishonest}, and no player will choose the strictly dominated strategy {honest}, therefore, this also leads both players into the prisoner's dilemma and no player can reconstruct the secret correctly.

We describe such a prisoner's dilemma through the game tree in Figure 2.

In this game tree,

- the players set is $P = \{p_1, p_2\}$ and the action set is $A = \{\text{honest}, \text{dishonest}\}$;
- the non-terminal selection node set is $H = \{n_0, n_1, n_2\}$ and the terminal node set is $E = \{n_3, n_4, n_5, n_6\}$;
- the information set is $I = \{I_1, I_2\}$ where $I_1 = \{n_0\}, I_2 = \{n_1,n_2\}$;
- the optional action set is $Q = \{Q_1,Q_2\}$ where $Q_1 = Q_2 = \{\text{honest}, \text{dishonest}\}$.

Lemma 1. The Game1 in Figure 2 has a sequential equilibrium $(s, \beta) = ((s_1, s_2), (\beta_1, \beta_2))$ where

$$\begin{cases} s_1 = ([0(\text{honest}), 1(\text{dishonest})]), \\ s_2 = ([0(\text{honest}), 1(\text{dishonest})]), \\ \beta_1 = ([1(n_0)]), \\ \beta_2 = ([0(n_1), 1(n_2)]). \end{cases}$$

Proof. We set the strategy profile in the sequential equilibrium is $s = (s_1, s_2)$ where

$$\begin{cases} s_1 = ([\rho_1(\text{honest}), \rho_2(\text{dishonest})]), \\ s_2 = ([\lambda_1(\text{honest}), \lambda_2(\text{dishonest})]). \end{cases}$$

Note that ρ_i, λ are probabilities and they satisfy

$$\begin{cases} \rho_i, \lambda_i \in [0, 1], \\ \rho_1 + \rho_2 = 1, \\ \lambda_1 + \lambda_2 = 1. \end{cases}$$

Then we derive belief system $\beta = (\beta_1, \beta_2)$ from Bayes' rule:

$$\begin{cases} \beta_1 = ([1(n_0)]), \\ \beta_2 = ([\rho_1(n_1), \rho_2(n_2)]). \end{cases}$$

Now we begin to prove the sequential rationality.

When p_2 reaches I_2, his/her expected utility is

$$u_2(s, \beta_2, I_2) = \beta_2(n_1) \cdot u_2(s, n_1) + \beta_2(n_2) \cdot u_2(s, n_2) = \rho_1 \cdot u_2(s, n_1) + \rho_2 \cdot u_2(s, n_2),$$

and we have

$$\begin{cases} u_2(s, n_1) = \lambda_1 \cdot u_2(n_3) + \lambda_2 \cdot u_2(n_4), \\ u_2(s, n_2) = \lambda_1 \cdot u_2(n_5) + \lambda_2 \cdot u_2(n_6). \end{cases}$$

Because

$$\begin{cases} u_2(n_3) = b, \\ u_2(n_4) = a, \\ u_2(n_5) = d, \\ u_2(n_6) = c, \end{cases}$$

and $a > b > c > d$, we can know $\lambda_1 = 0, \lambda_2 = 1$. To be more precise, p_2 must choose $\{\text{dishonest}\}$ in order to maximize his/her own utility.

Now let us consider p_1's situation. When he/she reaches I_1, his/her expected utility is

$$u_1(s, \beta_1, I_1) = 1(n_0) \cdot u_1(s, n_0) = \rho_1 \cdot \lambda_1 \cdot u_1(n_3) + \rho_1 \cdot \lambda_2 \cdot u_1(n_4) + \rho_2 \cdot \lambda_1 \cdot u_1(n_5) + \rho_2 \cdot \lambda_2 \cdot u_1(n_6).$$

Because rational p_1 also knows that p_2 must choose $\{\text{dishonest}\}$, i.e., $\lambda_1 = 0, \lambda_2 = 1$,

$$u_1(s, \beta_1, I_1) = \rho_1 \cdot \lambda_2 \cdot u_1(n_4) + \rho_2 \cdot \lambda_2 \cdot u_1(n_6).$$

Because

$$\begin{cases} u_1(n_4) = d, \\ u_1(n_6) = c, \end{cases}$$

and $c > d$, we can get that $\rho_1 = 0, \rho_2 = 1$.

Then we will prove the sequential consistency. Let the fully mixed $s^k = (s_1^k, s_2^k)$ where

$$\begin{cases} s_1^k = \left(\left[\dfrac{1}{k}(\text{honest}), \dfrac{k-1}{k}(\text{dishonest})\right]\right), \\ s_2^k = \left(\left[\dfrac{1}{k}(\text{honest}), \dfrac{k-1}{k}(\text{dishonest})\right]\right), \end{cases}$$

and the derived $\beta^k = (\beta_1^k, \beta_2^k)$ from Bayes rule is

$$\begin{cases} \beta_1^k = ([1(n_0)]), \\ \beta_2^k = \left(\left[\dfrac{1}{k}(n_1), \dfrac{k-1}{k}(n_2)\right]\right). \end{cases}$$

Because

$$\begin{cases} \lim\limits_{k\to\infty}\left(\dfrac{k-1}{k}\right) = 1, \\ \lim\limits_{k\to\infty}\left(\dfrac{1}{k}\right) = 0, \end{cases}$$

we can get the result

$$\begin{cases} \lim\limits_{k\to\infty}\left(s^k\right) \to s, \\ \lim\limits_{k\to\infty}\left(\beta^k\right) \to \beta. \end{cases}$$

Theorem 1. The Game1 in Figure 2 will always end at n_6.

Proof. From Lemma 1, we can know Game1 has unique sequential equilibrium, and RPath(n_6) means there is no motivation for rational players to be honest when they share secret shares, so Game1 will always begin with n_0, pass through n_2 and end at n_6.

We mark RPath(n_6) with the red line in Figure 2. Obviously, in this case, the secret cannot be reconstructed correctly and this is why we urgently need a solution to break the prisoner's dilemma.

4 Incentive contract

In order to solve the problems in Section 3, we redesign the process of secret sharing and propose an incentive contract (IC).

4.1 New system model

Our new system model contains not only a SGC and SSPs, but also a verifier V. The task of V is to verify the honesty of SSPs if the verification request is initiated.

Then we divides rational secret sharing into the following phases.

Distribution phase. The SGC

(1) randomly generates the shared key $W \in \{0,1\}^k$, and then it randomly selects w_1,$w_2 \in \{0,1\}^k$ and satisfies $W = w_1 \oplus w_2$;

(2) computes hash function $m_1 = h(w_1), m_2 = h(w_2)$;

(3) randomly selects $s_1, s_2 \in F_q^*$, and then it generates commitments $\text{Com}_{s_1}(m_1), \text{Com}_{s_2}(m_2)$;

(4) sends (w_1, s_1, h) to p_1 and sends (w_2, s_2, h) to p_2. Besides, it sends $(W, w_1, w_2, \text{Com}_{s_1}(m_1), \text{Com}_{s_2}(m_2), h)$ to V, and then it goes offline.

SGC will go offline after generating and distributing secret shares, so it needs to send useful messages to V in order to ensure verifiability.

Sign contract phase (SCP).

(1) V puts the commitments $\text{Com}_{s_1}(m_1), \text{Com}_{s_2}(m_2)$ as parts of the incentive contract and puts the contract on the blockchain.

(2) SSPs verify the source of contract by opening the commitments and choose whether to sign the contract or not.

For example, in the SCP, if p_i successfully opens $\mathrm{Com}_{s_1}(m_1)$ by s_1 and he/she gets $m_1 = h(w_1)$, then he/she will be able to believe that the contract is from V.

Publish secret share phase (PSSP).

(1) p_i computes hash function $m'_i = h(w'_i)$.

(2) p_i randomly selects $s'_i \in F_q^*$, and then he/she generates a commitment $\mathrm{Com}_{s'_i}(m'_i)$.

(3) p_i puts his/her commitment $\mathrm{Com}_{s'_i}(m'_i)$ on the blockchain, and sends (s'_i, w'_i) to the other player.

It means that in the PSSP, p_i needs to generate a commitment to the message he/she is sending to the other player. For example, if p_1 sends w'_1 to p_2, he/she needs to generate a commitment $\mathrm{Com}_{s'_1}(m'_1)$. Therefore, p_2 can verify that the commitment $\mathrm{Com}_{s'_1}(m'_1)$ on the blockchain is actually generated by the $h(w'_1)$ of the received w'_1. It also ensures that players cannot deny.

Promulgate reconstructed secret phase (PRSP).

(1) p_i verifies that whether the other player's commitment on the blockchain is exactly the result of computing a commitment for the message (s'_{-i}, m'_{-i}).

(2) p_i computes $W_i = w_i \oplus w'_{-i}$.

(3) p_i computes hash function $M'_i = h(W'_i)$.

(4) p_i randomly selects $S_i{}' \in F_q^*$, and then he/she generates a commitment $\mathrm{Com}_{S'_i}(M'_i)$.

(5) p_i puts his/her commitment $\mathrm{Com}_{S'_i}(M'_i)$ on the blockchain, and sends (S'_i, W'_i) to the other player.

Similarly, for example, in the PRSP, p_1 needs to commit to M'_1 by choosing a secret S'_1 to generate a commitment $\mathrm{Com}_{S'_1}(M'_1)$ and put $\mathrm{Com}_{S'_1}(M'_1)$ on the blockchain. Therefore, p_2 can verify that the commitment $\mathrm{Com}_{S'_1}(M'_1)$ on the blockchain is actually generated by the $h(W'_1)$ of the received W'_1. Besides, p_2 can judge p_1's honesty by comparing $h(W'_1)$ with $h(W_2)$ where $W_2 = w'_1 \oplus w_2$.

Verification phase.

Once a verification request is initiated, p_i must send (s'_i, w'_i, S'_i, W'_i) to V, and then V gradually verifies p_i's honesty by continually opening commitments. The detailed verification process will be described in the following Contract content.

Note that this phase begins only when a verification request is initiated. We call the player who first initiates the verification request p_{ldr} while the other player p_{flr}. Whether p_i sends the wrong message or refuses to send the message is easy to verify, so rational players must send the real (s'_i, S'_i) to V to avoid being confiscated deposit $r + g$.

4.2 Value variables

Since the incentive contract is introduced, we need to define more value variables.

- $r + g$: The deposit that players need to pay when signing incentive contract.
- r : Deposit to be confiscated if caught dishonest.
- g : Expense to be paid for initiating a verification request.
- v : Verification cost of verifier.
- ε : Some invisible utility for V when he/she signs the contract with players, such as earning a good reputation.

Obviously, $g - v > 0$, otherwise, V does not accept a verification request.

4.3 Contract content and analysis

We will introduce the specific content of IC, and then we analyze each step in detail by studying the utility function of rational players.

We define some deadlines in the contract, which help us to eliminate some strictly dominated strategies through utility function, thus simplifying the game tree.

- T_1 : In the SCP, the deadline for signing the contract.
- T_2 : In the PSSP, the deadline for sending secret share.
- T_3 : In the PRSP, the deadline for sending reconstructed secret.

Contract content.

- Step1: In the SCP, if p_1, p_2 both sign IC before T_1, IC will take effect and enter Step2; else p_1, p_2 will enter Game1.
- Step2: In the PSSP, if p_1, p_2 both send secret shares before T_2, IC will enter Step3; else the deposit $r + g$ of the silent player p_i will be confiscated and IC will terminate.

- Step3: In the PRSP, if p_1, p_2 both send reconstructed secrets before T_3, IC will enter Step4; else the deposit $r+g$ of the concealed player p_i will be confiscated and IC will terminate.
- Step4: If a player p_{ldr} initiates a verification request, IC will enter Step5; else the deposit $r+g$ of the two players will be returned and IC will enter Step6.
- Step5 (Verification phase): V verifies W'_{flr} and W'_{ldr}. If $W'_{ldr} = W'_{flr} = W$, IC will enter Step5.1; else if $W'_{flr} = W$, IC will enter Step5.2; else if $W'_{ldr} = W$, IC will enter Step5.3; else IC will enter Step5.4.
- Step5.1: Charging p_{ldr}'s deposit g for verification, returning deposit r of p_{ldr} and deposit $r+g$ of p_{flr}, then V sends W to p_{ldr} and IC terminates.
- Step5.2: V verifies whether w'_{flr} is wrong or not. If w'_{flr} is wrong, IC will enter Step5.2.1; else IC will enter Step5.2.2.
- Step5.2.1: Charging p_{ldr}'s deposit g for verification, transferring p_{flr}'s deposit r to p_{ldr}, returning p_{ldr}'s deposit r and p_{flr}'s deposit g, then V sends W to p_{ldr} and IC terminates.
- Step5.2.2: Charging p_{ldr}'s deposit g for verification, transferring p_{ldr}'s deposit r to p_{flr}, returning p_{flr}'s deposit $r+g$, then IC terminates.
- Step5.3: V verifies whether w'_{ldr} is wrong or not. If w'_{ldr} is wrong, IC will enter Step5.3.1; else IC will enter Step5.3.2.
- Step5.3.1: Charging p_{ldr}'s deposit g for verification, transferring p_{ldr}'s deposit r to p_{flr}, returning p_{flr}'s deposit $r+g$, then IC terminates.
- Step5.3.2: Charging p_{ldr}'s deposit g for verification, transferring p_{flr}'s deposit r to p_{ldr}, returning p_{ldr}'s deposit r and p_{flr}'s deposit g, then V sends W to p_{ldr} and IC terminates.
- Step5.4: V verifies w'_{ldr}, w'_{flr}. If $(((w'_{ldr} \neq w_{ldr})||(W_{ldr} \neq (w_{ldr} \oplus w'_{flr})))\&\&((w'_{flr} \neq w_{flr})||(W_{flr} \neq (w'_{ldr} \oplus w_{flr}))) = 1$, IC will enter Step5.4.1; else if $((w'_{ldr} \neq w_{ldr})||(W_{ldr} \neq (w_{ldr} \oplus w'_{flr}))) = 1$, IC will enter Step5.4.2; else if $((w'_{flr} \neq w_{flr})\&\&(W_{flr} \neq (w'_{ldr} \oplus w_{flr}))) = 0$, IC will enter Step5.4.3; else IC will enter Step5.4.4.
- Step5.4.1: Charging p_{ldr}'s deposit g for verification, confiscating p_{ldr}'s and p_{flr}'s deposit r, returning p_{flr}'s deposit g, then IC terminates.
- Step5.4.2: Charging p_{ldr}'s deposit g for verification, transferring p_{ldr}'s deposit r to p_{flr}, returning p_{flr}'s deposit $r+g$, then IC terminates.
- Step5.4.3: Charging p_{ldr}'s deposit g for verification, transferring p_{flr}'s deposit r to p_{ldr}, returning p_{ldr}'s deposit r and p_{flr}'s deposit g, then V sends W to p_{ldr} and IC terminates.
- Step5.4.4: Charging p_{ldr}'s deposit g for verification, confiscating p_{flr}'s deposit g, transferring p_{flr}'s deposit r to p_{ldr}, returning p_{ldr}'s deposit r, then V sends W to p_{ldr} and IC terminates.
- Step6: V does not verify and IC terminates.

When a verification request is initiated, V will charge p_{ldr}'s verification expense g. Then we will punish the dishonest player and transfer the deposit r to the honest player (both players' deposit r will be confiscated by V if there is no honest player). In addition, if p_{ldr} is honest, V will send the correct W to him/her after the verification.

Our analysis process is as follows.

Contract analysis.

* Step1 means that in the SCP, p_1, p_2 must sign IC to start it, otherwise the prisoners' dilemma will occur and both p_1, p_2 will enter Game1.

* Step2 means that in the PSSP, if a player chooses strategy {silent}, then his/her all deposit will be confiscated, and it is impossible for him/her to get worst utility. Therefore, the strategy {silent} is a strictly dominated strategy and no player has motivation to choose it.

* Step3 means that in the PRSP, the strategy {concealed} is a strictly dominated strategy and no player has motivation to choose it through a similar analysis with Step2.

* Step4 represents whether the players choose to verify or not. Note that player p_i can compare the secret W_i reconstructed by himself/herself and the secret W'_{-i} sent by the other player to judge each other's honesty.

* Step5 represents the verification process of V.

* Step5.1 means that both players are honest at this time, p_{ldr} is obviously "making trouble unreasonably", and initiating verification will only reduce his/her own utility:

$$u_{5.1}(p_{ldr}, p_{flr}) = (b-g, b).$$

* Step5.2 means that either p_{flr} cheats in the PSSP or p_{ldr} deceives in the PRSP.

* Step5.2.1 means that p_{flr} cheats in the PSSP, and

$$u_{5.2.1}(p_{ldr}, p_{flr}) = (b + r - g, b - r).$$

* Step5.2.2 means that p_{ldr} deceives in the PRSP, and

$$u_{5.2.2}(p_{ldr}, p_{flr}) = (b - r - g, b + r).$$

* Step5.3 means that either p_{ldr} cheats in the PSSP or p_{flr} deceives in the PRSP.

* Step5.3.1 means that p_{ldr} cheats in the PSSP, and

$$u_{5.3.1}(p_{ldr}, p_{flr}) = (a - r - g, d + r).$$

* Step5.3.2 means that p_{flr} deceives in the PRSP, and

$$u_{5.3.2}(p_{ldr}, p_{flr}) = (b + r - g, b - r).$$

* Step5.4 means that at least one player in p_{ldr}, p_{flr} is dishonest and needs to verify both w'_{ldr} and w'_{flr}.

* Step5.4.1 means that both p_{ldr}, p_{flr} are dishonest, and

(1) if they both reconstruct the secret correctly,

$$u_{5.4.1}(p_{ldr}, p_{flr}) = (b - r - g, b - r);$$

(2) else if only p_{ldr} reconstructs the secret correctly,

$$u_{5.4.1}(p_{ldr}, p_{flr}) = (a - r - g, d - r);$$

(3) else if only p_{flr} reconstructs the secret correctly,

$$u_{5.4.1}(p_{ldr}, p_{flr}) = (d - r - g, a - r);$$

(4) else

$$u_{5.4.1}(p_{ldr}, p_{flr}) = (c - r - g, c - r).$$

* Step5.4.2 means that only p_{ldr} is dishonest, and

(1) if they both reconstruct the secret correctly,

$$u_{5.4.2}(p_{ldr}, p_{flr}) = (b - r - g, b + r);$$

(2) else

$$u_{5.4.2}(p_{ldr}, p_{flr}) = (a - r - g, d + r).$$

* Step5.4.3 means that only p_{flr} is dishonest, and he/she either cheats in the PSSP or deceives in the PRSP, and we have

$$u_{5.4.3}(p_{ldr}, p_{flr}) = (b + r - g, b - r).$$

* Step5.4.4 means that only p_{flr} is dishonest, and he/she both cheats in the PSSP and deceives in the PRSP, and we have

$$u_{5.4.3}(p_{ldr}, p_{flr}) = (b + r - g, b - r - g).$$

* Step6 means that no player initiates a verification request, so the strategy in the PRSP does not affect the utility function. Therefore, in the PSSP,

(1) if both p_1, p_2 choose strategy {publish}, then

$$u(p_1, p_2) = (b, b);$$

(2) else if both p_1, p_2 choose strategy {cheat}, then

$$u(p_1, p_2) = (c, c);$$

Table 1 The p_{ldr}'s utility of IVR and NIVR

Serial	p_{ldr}	p_{ldr}	p_{flr}	p_{flr}	$u_{\mathrm{IVR}}(p_{ldr})$	$u_{\mathrm{NIVR}}(p_{ldr})$
1*	publish	promulgate	publish	promulgate	$b-g$	b
2*	publish	promulgate	publish	deceive	$b+r-g$	b
3*	publish	promulgate	cheat	promulgate	$b+r-g$	d
4*	publish	promulgate	cheat	deceive	$b+r-g$	d
5*	publish	deceive	publish	deceive	$b-r-g$	b
6*	publish	deceive	cheat	promulgate	$d-r-g$	d
7*	publish	deceive	cheat	deceive	$d-r-g$	d
8*	cheat	promulgate	publish	deceive	$a-r-g$	a
9*	cheat	promulgate	cheat	promulgate	$c-r-g$	c
10*	cheat	promulgate	cheat	deceive	$c-r-g$	c
11*	cheat	deceive	publish	deceive	$a-r-g$	a
12*	cheat	deceive	cheat	promulgate	$c-r-g$	c
13*	cheat	deceive	cheat	deceive	$c-r-g$	c
14*	publish	deceive	publish	promulgate	$b-r-g$	b
15*	cheat	promulgate	publish	promulgate	$a-r-g$	a
16*	cheat	deceive	publish	promulgate	$a-r-g$	a

(3) else

$$u(p_{\mathrm{cheat}}, p_{\mathrm{publish}}) = (a, d).$$

Through the analysis of IC, we can know when $r > g$, players' choice of whether to initiate a verification request directly affects the utility function. And we draw Table 1 to show the p_{ldr}'s utility of initiating a verification request (IVR) and not initiating a verification request (NIVR).

Lemma 2. Honest player has no motivation to initiate a verification request if the other is as well as honest.

Proof. Serial 1* in Table 1 means both p_{ldr} and p_{flr} are honest, in this case, we have

$$u_{\mathrm{IVR}}(p_{ldr}) < u_{\mathrm{NIVR}}(p_{ldr}).$$

Lemma 3. If $r - g > 0$, honest player inevitably initiates a verification request if the other is dishonest.

Proof. Serials 2*–4* in Table 1 mean p_{ldr} is honest while p_{flr} is dishonest, in this case, because $r - g > 0$, so we have

$$u_{\mathrm{IVR}}(p_{ldr}) > u_{\mathrm{NIVR}}(p_{ldr}).$$

Lemma 4. Dishonest player must not dare to initiate a verification request.

Proof. Serials 5*–16* in Table 1 mean p_{ldr} is dishonest, in this case, we have

$$u_{\mathrm{IVR}}(p_{ldr}) < u_{\mathrm{NIVR}}(p_{ldr}).$$

4.4 Game and analysis

From the previous analysis, we can know that {silent} in the PSSP and {concealed} in the PRSP are strictly dominated strategies, so {silent}, {concealed} are directly eliminated. Then after the PRSP, rational players have reason to choose whether to initiate a verification request or not according to different situations, reasonable explanations and proofs will be given in the following Theorem 2. For these reasons, our extensive game tree is simplified and constructed.

- The player set is $P = \{p_1, p_2\}$, non-terminal selection node set is $H = \{n_0, n_1, \ldots, n_{16}\}$, terminal node set is $E = \{n_{17}, n_{18}, \ldots, n_{32}\}$.
- {silent} and {concealed} are strictly dominated strategies, so these strategies can be directly removed from the game tree, and the action set is A ={sign, no_sign, publish, cheat, promulgate, deceive}.
- The information set is $I = \{I_{11}, I_{12}, I_{13}, I_{14}, I_{21}, I_{22}, I_{23}, I_{24}\}$ where $I_{11} = \{n_0\}, I_{12} = \{n_2\}, I_{13} = \{n_5, n_6\}, I_{14} = \{n_7, n_8\}$ and $I_{21} = \{n_1\}, I_{22} = \{n_3, n_4\}, I_{23} = \{n_9, n_{10}, n_{13}, n_{14}\}, I_{24} = \{n_{11}, n_{12}, n_{15}, n_{16}\}$.
- The optional action set is $Q = \{Q_{11}, Q_{12}, Q_{13}, Q_{14}, Q_{21}, Q_{22}, Q_{23}, Q_{24}\}$ where $Q_{11} = Q_{21} =$ {sign, no_sign}, $Q_{12} = Q_{22} =$ {publish, cheat} and $Q_{13} = Q_{14} = Q_{23} = Q_{24} =$ {promulgate, deceive}.

Table 2 Utility function in Game2

Node	$u(p_1)$	$u(p_2)$
n_{17}	b	b
n_{18}	$b+r-g$	$b-r$
n_{19}	$b-r$	$b+r-g$
n_{20}	b	b
n_{21}	$b+r-g$	$b-r$
n_{22}	$b+r-g$	$b-r-g$
n_{23}	d	a
n_{24}	d	a
n_{25}	$b-r$	$b+r-g$
n_{26}	a	d
n_{27}	$b-r-g$	$b+r-g$
n_{28}	a	d
n_{29}	c	c
n_{30}	c	c
n_{31}	c	c
n_{32}	c	c
Game1	c	c

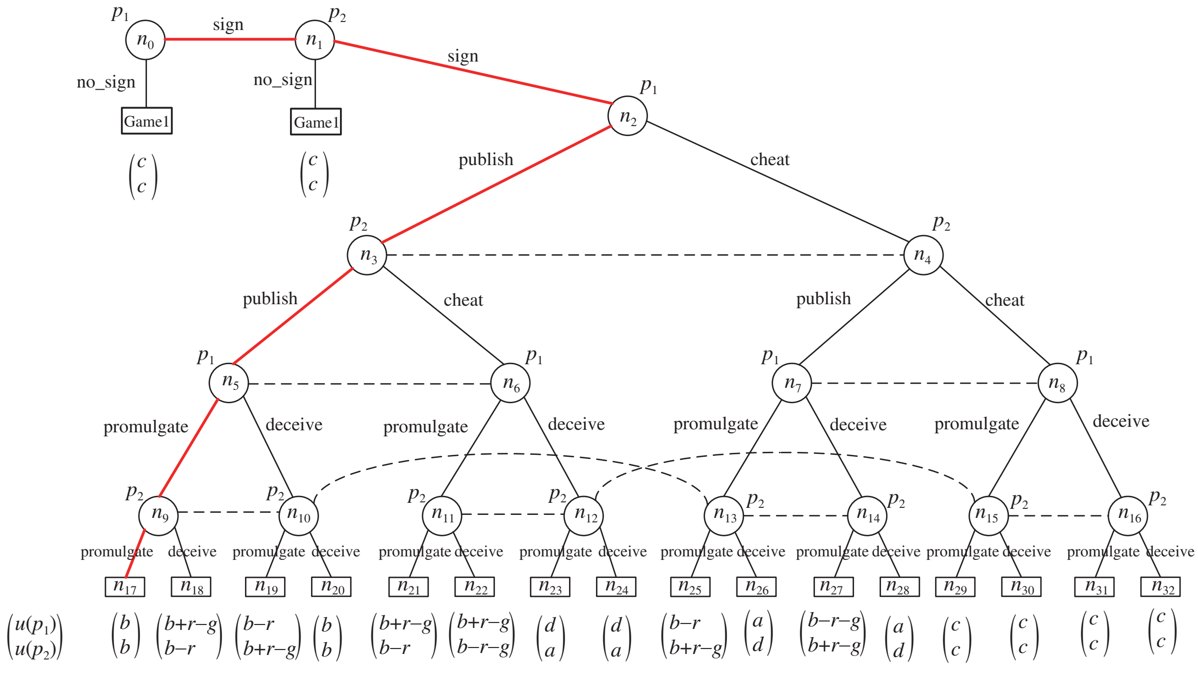

Figure 3 Game2.

Theorem 2. In Game2, we can assign reasonable probabilities for players to IVR and NIVR in different situations after the PRSP, thus removing unnecessary branches of the game tree.

Proof. From Lemma 2, we can see that the path ending at node n_{17} means because both players are honest, no one will initiate a verification request. From Lemma 3, the paths ending at node $\{n_{18}, n_{19}, n_{21}, n_{22}, n_{25}, n_{27}\}$ mean because one player is honest while the other is dishonest, the honest player will inevitably initiate a verification request. From Lemma 4, the paths ending at node $\{n_{20}, n_{23}, n_{24}, n_{26}, n_{28}, n_{29}, n_{30}, n_{31}, n_{32}\}$ mean because both players are dishonest, they will not dare to initiate a verification request.

Now we draw Table 2 to facilitate our subsequent proof.

Lemma 5. If $r > g$, Game2 in Figure 3 exists a sequential equilibrium $(s, \beta) = ((s_1, s_2), (\beta_1, \beta_2))$

where

$$\begin{cases} s_1=([1(\text{sign}),0(\text{no\_sign})],[1(\text{publish}),0(\text{cheat})],[1(\text{promulgate}),0(\text{deceive})],[1(\text{promulgate}),0(\text{deceive})]),\\ s_2=([1(\text{sign}),0(\text{no\_sign})],[1(\text{publish}),0(\text{cheat})],[1(\text{promulgate}),0(\text{deceive})],[1(\text{promulgate}),0(\text{deceive})]),\\ \beta_1=([1(n_0)],[1(n_2),0(\text{Game1})],[1(n_5),0(n_6)],[1(n_7),0(n_8)]),\\ \beta_2=([1(n_1),0(\text{Game1})],[1(n_3),0(n_4)],[1(n_9),0(n_{10}),0(n_{13}),0(n_{14})],[1(n_{11}),0(n_{12}),0(n_{15}),0(n_{16})]). \end{cases}$$

Proof. We set the strategy profile in the sequential equilibrium is $s=(s_1,s_2)$ where

$$\begin{cases} s_1=([\rho_1(\text{sign}),\rho_2(\text{no\_sign})],[\rho_3(\text{publish}),\rho_4(\text{cheat})],[\rho_5(\text{promulgate}),\rho_6(\text{deceive})],\\ \qquad [\rho_7(\text{promulgate}),\rho_8(\text{deceive})]),\\ s_2=([\lambda_1(\text{sign}),\lambda_2(\text{no\_sign})],[\lambda_3(\text{publish}),\lambda_4(\text{cheat})],[\lambda_5(\text{promulgate}),\lambda_6(\text{deceive})],\\ \qquad [\lambda_7(\text{promulgate}),\lambda_8(\text{deceive})]), \end{cases}$$

ρ_i,λ_i are probabilities and they satisfy

$$\begin{cases} \rho_i,\lambda_i\in[0,1],\\ \rho_1+\rho_2=1,\ \rho_3+\rho_4=1,\ \rho_5+\rho_6=1,\ \rho_7+\rho_8=1,\\ \lambda_1+\lambda_2=1,\ \lambda_3+\lambda_4=1,\ \lambda_5+\lambda_6=1,\ \lambda_7+\lambda_8=1. \end{cases}$$

Then we derive belief system $\beta=(\beta_1,\beta_2)$ from Bayes' rule,

$$\begin{cases} \beta_1=([1(n_0)],[\lambda_1(n_2),\lambda_2(\text{Game1})],[\lambda_3(n_5),\lambda_4(n_6)],[\lambda_5(n_7),\lambda_6(n_8)]),\\ \beta_2=([\rho_1(n_1),\rho_2(\text{Game1})],[\rho_3(n_3),\rho_4(n_4)],[\rho_3\cdot\rho_5(n_9),\rho_3\cdot\rho_6(n_{10}),\rho_4\cdot\rho_7(n_{13}),\rho_4\cdot\rho_8(n_{14})],\\ \qquad [\rho_3\cdot\rho_5(n_{11}),\rho_3\cdot\rho_6(n_{12}),\rho_4\cdot\rho_7(n_{15}),\rho_4\cdot\rho_8(n_{16})]). \end{cases}$$

Now we will first prove the sequential rationality (see (a)–(h)).

(a) When p_2 reaches information set $I_{23}=\{n_9,n_{10},n_{13},n_{14}\}$, the expected utility of player p_2 is

$$\begin{aligned} u_2(s,\beta_2,I_{23}) &= \beta_2(n_9)\cdot u_2(s,n_9)+\beta_2(n_{10})\cdot u_2(s,n_{10})+\beta_2(n_{13})\cdot u_2(s,n_{13})+\beta_2(n_{14})\cdot u_2(s,n_{14})\\ &= \rho_3\cdot\rho_5\cdot u_2(s,n_9)+\rho_3\cdot\rho_6\cdot u_2(s,n_{10})+\rho_4\cdot\rho_7\cdot u_2(s,n_{13})+\rho_4\cdot\rho_8\cdot u_2(s,n_{14}), \end{aligned}$$

and we have

$$\begin{cases} u_2(s,n_9)=\lambda_5\cdot u_2(n_{17})+\lambda_6\cdot u_2(n_{18}),\\ u_2(s,n_{10})=\lambda_5\cdot u_2(n_{19})+\lambda_6\cdot u_2(n_{20}),\\ u_2(s,n_{13})=\lambda_5\cdot u_2(n_{25})+\lambda_6\cdot u_2(n_{26}),\\ u_2(s,n_{14})=\lambda_5\cdot u_2(n_{27})+\lambda_6\cdot u_2(n_{28}). \end{cases}$$

If $r-g>0$, we can easily get

$$u_2(n_{17})>u_2(n_{18}),\quad u_2(n_{19})>u_2(n_{20}),\quad u_2(n_{25})>u_2(n_{26}),\quad u_2(n_{27})>u_2(n_{28}).$$

Therefore, we can infer that $\lambda_5=1$ and $\lambda_6=0$ is reasonable because it can maximize p_2's own utility. It means that when p_2 reaches I_{23}, he/she must choose {promulgate} to maximize his/her own utility.

(b) When p_2 reaches information set $I_{24}=\{n_{11},n_{12},n_{15},n_{16}\}$, the expected utility of player p_2 is

$$\begin{aligned} u_2(s,\beta_2,I_{24}) &= \beta_2(n_{11})\cdot u_2(s,n_{11})+\beta_2(n_{12})\cdot u_2(s,n_{12})+\beta_2(n_{15})\cdot u_2(s,n_{15})+\beta_2(n_{16})\cdot u_2(s,n_{16})\\ &= \rho_3\cdot\rho_5\cdot u_2(s,n_{11})+\rho_3\cdot\rho_6\cdot u_2(s,n_{12})+\rho_4\cdot\rho_7\cdot u_2(s,n_{15})+\rho_4\cdot\rho_8\cdot u_2(s,n_{16}), \end{aligned}$$

and we have

$$\begin{cases} u_2(s,n_{11})=\lambda_7\cdot u_2(n_{21})+\lambda_8\cdot u_2(n_{22}),\\ u_2(s,n_{12})=\lambda_7\cdot u_2(n_{23})+\lambda_8\cdot u_2(n_{24}),\\ u_2(s,n_{15})=\lambda_7\cdot u_2(n_{29})+\lambda_8\cdot u_2(n_{30}),\\ u_2(s,n_{16})=\lambda_7\cdot u_2(n_{31})+\lambda_8\cdot u_2(n_{32}). \end{cases}$$

We can easily get

$$u_2(n_{21}) > u_2(n_{22}), \quad u_2(n_{23}) = u_2(n_{24}), \quad u_2(n_{29}) = u_2(n_{30}), \quad u_2(n_{31}) = u_2(n_{32}).$$

Therefore, we can infer that $\lambda_7 = 1$ and $\lambda_8 = 0$. It means that when p_2 reaches I_{24}, he/she must choose {promulgate} to maximize his/her own utility.

So from (a) and (b), we can know that in the PRSP, p_2 must choose {promulgate} in order to maximize his/her own utility.

(c) Let us continue our analysis. When p_1 reaches information set $I_{13} = \{n_5, n_6\}$, the expected utility of player p_1 is

$$\begin{aligned} u_1(s, \beta_1, I_{13}) &= \beta_1(n_5) \cdot u_1(s, n_5) + \beta_1(n_6) \cdot u_1(s, n_6) \\ &= \lambda_3 \cdot u_1(s, n_5) + \lambda_4 \cdot u_1(s, n_6), \end{aligned}$$

and we have

$$\begin{cases} u_1(s, n_5) = \lambda_5 \cdot \rho_5 \cdot u_1(n_{17}) + \lambda_6 \cdot \rho_5 \cdot u_1(n_{18}) + \lambda_5 \cdot \rho_5 \cdot u_1(n_{19}) + \lambda_6 \cdot \rho_6 \cdot u_1(n_{20}), \\ u_1(s, n_6) = \lambda_7 \cdot \rho_5 \cdot u_1(n_{21}) + \lambda_8 \cdot \rho_5 \cdot u_1(n_{22}) + \lambda_7 \cdot \rho_5 \cdot u_1(n_{23}) + \lambda_8 \cdot \rho_6 \cdot u_1(n_{24}). \end{cases}$$

Because not only does p_2 know that {promulgate} must be chosen to maximize p_2's utility in the PRSP, but p_1 also knows that. In other words, p_1 also knows that

$$\begin{cases} \lambda_5 = 1, \ \lambda_6 = 0, \\ \lambda_7 = 1, \ \lambda_8 = 0, \end{cases}$$

so we have

$$\begin{cases} u_1(s, n_5) = \rho_5 \cdot u_1(n_{17}) + \rho_6 \cdot u_1(n_{19}), \\ u_1(s, n_6) = \rho_5 \cdot u_1(n_{21}) + \rho_6 \cdot u_1(n_{23}), \end{cases}$$

and we can easily get

$$u_1(n_{17}) > u_1(n_{19}), \quad u_1(n_{21}) > u_1(n_{23}).$$

Therefore, $\rho_5 = 1, \rho_6 = 0$ and it means p_1 must choose {promulgate} to maximize his/her own utility at information set I_{13}.

(d) When p_1 reaches information set $I_{14} = \{n_7, n_8\}$, the expected utility of player p_1 is

$$\begin{aligned} u_1(s, \beta_1, I_{14}) &= \beta_1(n_7) \cdot u_1(s, n_7) + \beta_1(n_8) \cdot u_1(s, n_8) \\ &= \lambda_5 \cdot u_1(s, n_7) + \lambda_6 \cdot u_1(s, n_8). \end{aligned}$$

Similarly, we have

$$\begin{cases} u_1(s, n_7) = \rho_7 \cdot u_1(n_{25}) + \rho_8 \cdot u_1(n_{27}), \\ u_1(s, n_8) = \rho_7 \cdot u_1(n_{29}) + \rho_8 \cdot u_1(n_{31}), \end{cases}$$

and we can easily get

$$u_1(n_{25}) > u_1(n_{27}), \quad u_1(n_{29}) = u_1(n_{31}).$$

Therefore, $\rho_7 = 1, \rho_8 = 0$ and it means p_1 must choose {promulgate} to maximize his/her own utility at information set I_{14}.

From (a)–(d), we can know p_1, p_2 must choose {promulgate} in the PRSP.

(e) When p_2 reaches I_{22}, the expected utility of player p_2 is

$$\begin{aligned} u_2(s, \beta_2, I_{22}) &= \beta_2(n_3) \cdot u_2(s, n_3) + \beta_2(n_4) \cdot u_2(s, n_4) \\ &= \rho_3 \cdot u_2(s, n_3) + \rho_4 \cdot u_2(s, n_4). \end{aligned}$$

Similarly, because p_2 knows p_1, p_2 must choose {promulgate} in the PRSP, i.e.,

$$\begin{cases} \lambda_5 = 1, \ \lambda_6 = 0, \ \lambda_7 = 1, \ \lambda_8 = 0, \\ \rho_5 = 1, \ \rho_6 = 0, \ \rho_7 = 1, \ \rho_8 = 0, \end{cases}$$

we have

$$\begin{cases} u_2(s,n_3)=\lambda_3\cdot u_2(n_{17})+\lambda_4\cdot u_2(n_{21}),\\ u_2(s,n_4)=\lambda_3\cdot u_2(n_{25})+\lambda_4\cdot u_2(n_{29}),\end{cases}$$

and we can easily get

$$u_2(n_{17})>u_2(n_{21}),\quad u_2(n_{25})>u_2(n_{29}).$$

Therefore, $\lambda_3=1,\lambda_4=0$ and it means p_2 must choose {publish} to maximize his/her own utility at information set I_{22}.

(f) When p_1 reaches I_{12}, the expected utility of player p_1 is

$$\begin{aligned} u_1(s,\beta_1,I_{12})&=\beta_1(n_2)\cdot u_1(s,n_2)+\beta_1(\text{Game1})\cdot u_1(s,\text{Game1})\\ &=\lambda_1\cdot u_1(s,n_2)+\lambda_2\cdot u_1(s,\text{Game1}).\end{aligned}$$

In particular, we should not confuse Game1 with the game after signing IC. Our rational players are fully aware of the validity of IC. If the IC does not come into effect, p_1,p_2 can only fall into prisoner's dilemma and get utility c. If IC comes into effect, p_1,p_2 will make independent strategic choices. In other words, the game after signing IC and Game1 are completely independent.

Similarly, p_1 knows

$$\begin{cases} \lambda_3=1,\ \lambda_4=0,\ \lambda_5=1,\ \lambda_6=0,\ \lambda_7=1,\ \lambda_8=0,\\ \rho_5=1,\ \rho_6=0,\ \rho_7=1,\ \rho_8=0.\end{cases}$$

In the above, we have

$$\begin{cases} u_1(s,n_2)=\rho_3\cdot u_1(n_{17})+\rho_4\cdot u_1(n_{25}),\\ u_1(s,\text{Game1})=c,\end{cases}$$

and we can easily get

$$u_1(n_{17})>u_1(n_{25}).$$

Therefore, $\rho_3=1,\rho_4=0$ and it means p_1 must choose {publish} to maximize his/her own utility at information set I_{12}.

(g) When p_2 reaches I_{21}, the expected utility of player p_2 is

$$\begin{aligned} u_2(s,\beta_2,I_{21})&=\beta_2(n_1)\cdot u_2(s,n_1)+\beta_2(\text{Game1})\cdot u_2(s,\text{Game1})\\ &=\rho_1\cdot u_2(s,n_1)+\rho_2\cdot u_2(s,\text{Game1}).\end{aligned}$$

p_2 knows

$$\begin{cases} \lambda_3=1,\ \lambda_4=0,\ \lambda_5=1,\ \lambda_6=0,\ \lambda_7=1,\lambda_8=0,\\ \rho_3=1,\ \rho_4=0,\ \rho_5=1,\ \rho_6=0,\ \rho_7=1,\ \rho_8=0,\end{cases}$$

and we have

$$\begin{cases} u_2(s,n_1)=\lambda_1\cdot u_2(n_{17})+\lambda_2\cdot c,\\ u_2(s,\text{Game1})=c.\end{cases}$$

We can easily get

$$u_2(n_{17})>c.$$

Therefore, $\lambda_1=1,\lambda_2=0$.

(h) When p_1 reaches I_{11}, the expected utility of player p_1 is

$$u_1(s,\beta_1,I_{11})=1(n_0)\cdot u_1(s,n_0).$$

p_2 knows

$$\begin{cases} \lambda_1=1,\ \lambda_2=0,\ \lambda_3=1,\ \lambda_4=0,\ \lambda_5=1,\lambda_6=0,\lambda_7=1,\lambda_8=0,\\ \rho_3=1,\ \rho_4=0,\ \rho_5=1,\ \rho_6=0,\ \rho_7=1,\ \rho_8=0,\end{cases}$$

so we have

$$u_1(s,n_0)=\rho_1\cdot u_1(n_{17})+\rho_2\cdot c,$$

and then we easily get

$$u_1(n_{17}) > c.$$

Therefore, $\rho_1 = 1, \rho_2 = 0$.

So from (a)–(h), we can get that

$$\begin{cases} \lambda_1 = 1,\ \lambda_2 = 0,\ \lambda_3 = 1,\ \lambda_4 = 0,\ \lambda_5 = 1, \lambda_6 = 0,\ \lambda_7 = 1,\ \lambda_8 = 0, \\ \rho_1 = 1,\ \rho_2 = 0,\ \rho_3 = 1,\ \rho_4 = 0,\ \rho_5 = 1,\ \rho_6 = 0,\ \rho_7 = 1,\ \rho_8 = 0, \end{cases}$$

and

$$\Rightarrow \begin{cases} \rho_3 \cdot \rho_5 = 1, \\ \rho_3 \cdot \rho_6 = 0, \\ \rho_4 \cdot \rho_7 = 0, \\ \rho_4 \cdot \rho_8 = 0. \end{cases}$$

In summary, $(s, \beta) = ((s_1, s_2), (\beta_1, \beta_2))$ satisfies the sequential rationality.

Then we will prove the sequential consistency (see (i)–(j)).

(i) We set the fully mixed sequence $s^k = (s_1^k, s_2^k)$ where

$$\begin{cases} s_1^k = \left(\left[\frac{k-1}{k}(\text{sign}), \frac{1}{k}(\text{no\_sign}) \right], \left[\frac{k-1}{k}(\text{publish}), \frac{1}{k}(\text{cheat}) \right], \right. \\ \qquad \left. \left[\frac{k-1}{k}(\text{promulgate}), \frac{1}{k}(\text{deceive}) \right], \left[\frac{k-1}{k}(\text{promulgate}), \frac{1}{k}(\text{deceive}) \right] \right), \\ s_2^k = \left(\left[\frac{k-1}{k}(\text{sign}), \frac{1}{k}(\text{no\_sign}) \right], \left[\frac{k-1}{k}(\text{publish}), \frac{1}{k}(\text{cheat}) \right], \right. \\ \qquad \left. \left[\frac{k-1}{k}(\text{promulgate}), \frac{1}{k}(\text{deceive}) \right], \left[\frac{k-1}{k}(\text{promulgate}), \frac{1}{k}(\text{deceive}) \right] \right). \end{cases}$$

Because

$$\begin{cases} \lim\limits_{k\to\infty} \left(\frac{k-1}{k} \right) = 1, \\ \lim\limits_{k\to\infty} \left(\frac{1}{k} \right) = 0, \end{cases}$$

we can get

$$\begin{cases} \lim\limits_{k\to\infty} (s_1^k) \to s_1 \\ \lim\limits_{k\to\infty} (s_2^k) \to s_2 \end{cases} \Rightarrow \lim_{k\to\infty} (s^k) \to s.$$

(j) Now we derive belief system $\beta^k = (\beta_1^k, \beta_2^k)$ through Bayes' rule where

$$\begin{cases} \beta_1^k = \left([1(n_0)], \left[\frac{k-1}{k}(n_2), \frac{1}{k}(\text{Game1}) \right], \left[\frac{k-1}{k}(n_5), \frac{1}{k}(n_6) \right], \left[\frac{k-1}{k}(n_7), \frac{1}{k}(n_8) \right] \right), \\ \beta_2^k = \left(\left[\frac{k-1}{k}(n_1), \frac{1}{k}(\text{Game1}) \right], \left[\frac{k-1}{k}(n_3), \frac{1}{k}(n_4) \right], \right. \\ \qquad \left[\frac{(k-1)^2}{k^2}(n_9), \frac{k-1}{k^2}(n_{10}), \frac{1}{k^2}(n_{13}), \frac{k-1}{k^2}(n_{14}) \right], \\ \qquad \left. \left[\frac{(k-1)^2}{k^2}(n_{11}), \frac{k-1}{k^2}(n_{12}), \frac{1}{k^2}(n_{15}), \frac{k-1}{k^2}(n_{16}) \right] \right). \end{cases}$$

Because

$$\begin{cases} \lim\limits_{k\to\infty}\left(\dfrac{k-1}{k}\right)=1, \\ \lim\limits_{k\to\infty}\left(\dfrac{1}{k}\right)=0, \\ \lim\limits_{k\to\infty}\dfrac{(k-1)^2}{k^2}=1, \\ \lim\limits_{k\to\infty}\dfrac{k-1}{k^2}=0, \\ \lim\limits_{k\to\infty}\dfrac{1}{k^2}=0, \end{cases}$$

we can get

$$\begin{cases} \lim\limits_{k\to\infty}(\beta_1^k)\to\beta_1 \\ \lim\limits_{k\to\infty}(\beta_2^k)\to\beta_2 \end{cases} \Rightarrow \lim_{k\to\infty}(\beta^k)\to\beta.$$

Theorem 3. If $r > g$, Game2 in Figure 3 will always end at n_{17}.

Proof. From Lemma 5 we can know that Game2 exists only one sequential equilibrium and we use the red line to mark the unique RPath(n_{17}). It means that rational players are confident that both of them will sign IC in the SCP, and then they honestly share secrets in the PSSP and PRSP. So Game2 begins at n_0, passes through $(n_1, n_2, n_3, n_5, n_9)$ and ends at n_{17}.

4.5 Verifier's motivation

In the previous article, we describe the motivation of p_1, p_2 to sign IC and be honest in the PSSP, PRSP. However, we do not involve the motivation of V to sign IC with p_1, p_2 in our analysis. Here we will explain it in detail.

V's verification cost is v, and once a verification request is initiated, V will charge p_{ldr}'s verification expense g. V gets the utility $v - g$ and because $v - g > 0$, V has no motivation to refuse a verification request.

However, from Theorem 3 we can see that p_1, p_2 are honest and have no motivation to initiate a verification request. Of course rational V knows that and the verification expense g cannot be obtained. So why V signs IC?

In fact, we think there will be some invisible utility ε for him/her such as earning a good reputation if V signs IC. Therefore, in order to obtain ε, V has the motivation to sign IC.

5 Simulation and analysis

The concept of programmable smart contracts can be traced back to [25], which is widely used with the introduction of the blockchain. In this section, we simulate the smart contract on the Ethereum, and then we analyze the requirements and overhead of the entire scheme.

5.1 Smart contract

Our smart contract mainly contains the functions in Table 3. For example, the function TimeOut is defined to avoid player's misbehavior, i.e., the player has no motivation to choose {silent} in the PSSP or {concealed} in the PRSP.

We know from Section 4 that there is a unique RPath(n_{17}) in the game and we simulate the execution of the rational path, i.e., the rational player p_i will

- call function Sign before T_1 and pay the deposit $r + g$.
- call function SendSecretShare before T_2 and send correct commitment $\mathrm{Com}_{s_i}(m_i)$.
- call function SendReconstructedSecret before T_3 and send the correct commitment $\mathrm{Com}_{S_i}(M_i)$.
- call function IVROrNot before T_4 and set the IVR = false, that is, not to initiate a verification request.

Table 3 Functions in the smart contract

Serial	Function	Description
1	Sign	A player calls this function to sign IC before T_1. Auto-transferring each player's deposit $r+g$ to the contract account if the contract takes effect.
2	SendSecretShare	A player calls this function to send the commitment of his/her secret share before T_2.
3	SendReconstructedSecret	A player calls this function to send the commitment of his/her reconstructed secret before T_3.
4	IVROrNot	A player calls this function to choose whether initiate a verification request or not before T_4.
5	Transfer	This function can handle disputes in different situations: If (UIVR) contract auto-transfers deposit $r+g$ to p_{ldr}, deposit $r+g$ to p_{flr}. Else if (IVR) Case1 (Both players are honest): contract auto-transfers deposit r to p_{ldr}, deposit $r+g$ to p_{flr} and deposit g to V; Case2 (p_{ldr} is honest while p_{flr} is dishonest): contract auto-transfers deposit $2r$ to p_{ldr}, deposit g to p_{flr} and deposit g to V; Case3 (p_{ldr} is dishonest while p_{flr} is honest): contract auto-transfers deposit $2r+g$ to p_{flr} and deposit g to V; Case4 (Both players are dishonest, p_{flr} either cheats in the PSSP or deceives in the PRSP): contract auto-transfers deposit g to p_{flr} and deposit $2r+g$ to V; Case5 (both players are dishonest, p_{flr} both cheats in the PSSP and deceives in the PRSP): contract auto-transfers deposit $2r+2g$ to V.
6	TimeOut	V can call this function if it times out: Case1 (Only p_i called the function SendSecretShare before T_2): contract auto-transfers deposit $r+g$ to p_i, deposit $r+g$ to V; Case2 (No player called the function SendSecretShare before T_2): contract auto-transfers deposit $2r+2g$ to V; Case3 (Only p_i called the function SendReconstructedSecret before T_3): contract auto-transfers deposit $r+g$ to p_i, deposit $r+g$ to V; Case4 (No player called the function SendReconstructedSecret before T_3): contract auto-transfers deposit $2r+2g$ to V; Case5 (Only p_i called the function IVROrNot before T_4): contract auto-transfers deposit $r+g$ to p_i, deposit $r+g$ to V; Case6 (No player called the function IVROrNot before T_4): contract auto-transfers deposit $2r+2g$ to V.

So we can get

- the function Transfer will return each player's deposit $r+g$,
- the function TimeOut will not be called.

5.2 Requirement

Our scheme mainly needs to consider the following requirements.

- Privacy: For players in secret sharing, their secrets need to be hidden. However, the data on the blockchain is visible to the public, so privacy should be considered when designing the smart contract.
- Verifiability: Only if the scheme satisfies verifiability can we guarantee that players will not deviate from honest behavior. In addition, rational players must be able to autonomously capture other's dishonesty to initiate a verification request.

In order to solve the above requirements, we use the famous Pedersen commitment scheme [26] and a collision-resistant hash function. More specifically, we use the hash function to compute the secrets such as $m_i' = h(w_i')$. Then, in the PSSP, p_i randomly generates $\mathrm{Com}_{s_i'}(m_i')$ by choosing random s_i' and sends $\mathrm{Com}_{s_i'}(m_i')$ to the blockchain, and in addition, p_i sends the randomly chosen s_i' and his/her secret shares w_i' to the p_{-i} through a secure channel. For a given $\mathrm{Com}_{s_i'}(m_i')$, because the commitment is hiding, so it is infeasible to know m_i' without s_i'. Besides, the commitment is binding; i.e., it is infeasible to find different pairs (s_i', m_i') and $(s_i'{}', {m_i}'')$ such that $\mathrm{Com}_{s_i'}(m_i') = \mathrm{Com}_{s_i''}({m_i}'')$. So when p_{-i} receives $(s_i', {w_i}')$, he/she computes $\mathrm{Com}_{s_i'}(h(w_i'))$, and then he/she can compare whether the two commitments (one is placed on the blockchain by p_i and the other is calculated by p_{-i}) are equal. Similarly, p_i commits reconstructed W_i in the PRSP. It is worth noting that if both players are honest, then the reconstructed secrets of both players should be consistent; i.e., if p_{-i} is honest, then he/she can judge the honesty of p_i by opening p_i's commitment $\mathrm{Com}_{S_i'}(M_i')$. Similarly, if a player p_{ldr} initiates a verification request, the verifier V also achieves verifiability by constantly opening commitment. Therefore, our scheme satisfies privacy and verifiability.

Table 4 Cost of using the smart contract

Serial	Function	Cost in Gas	Cost in Dollar
1	Deploy	1382691	0.26
2	Sign	292963	0.056
3	SendSecretShare	132224	0.025
4	SendReconstructedSecret	132180	0.025
5	IVROrNot	95416	0.018
6	Transfer	41748	0.008

Table 5 Number of calculations

Player	Generate commitment	Open commitment	Compute hash function
p_1	2	3	5
p_2	2	3	5
V	0	0	0
Total	4	6	10

5.3 Overhead

Cost is measured in terms of gas in the smart contract. The gas price is 1 Gas = 1 Gwei (1×10^{-9} ether) in all transactions, and the current exchange rate is 1 ether = \$190.45. Then we show the cost of deploying the contract and executing the functions in Table 4.

As we can see, the total cost of using the smart contract is low. The Deploy(\$0.26) is used to deploy the contract on the blockchain, and we find other functions cost less. Note that the player's calculation is done privately and not published on the blockchain; i.e., the player only needs to send the commitment.

In addition, we know from previous analysis that no rational player has dishonest motivation. So we show the actual number of calculations in Table 5, and we can get that additional overhead incurred by cryptography is also small.

6 Conclusion and future work

We consider rational secret sharing as a dynamic game with imperfect information. We find that rational players will never cooperate if the incentive mechanism is not effective enough, which leads to the fact that the correct secret cannot be reconstructed. The sequential equilibrium, as an refinement of Nash equilibrium, specifies sequential rationality and sequential consistency, which can even eliminate incredible threats in dynamic games with imperfect information. In this paper, we have constructed an incentive-compatible rational secret sharing scheme and found unique sequential equilibrium solution (in this equilibrium case no rational player has a motivation to deviate from honest behavior). What is more, we use blockchain and a smart contract to control the utility functions of rational players, which makes our scheme feasible in practice applications. In other words, each player must pay a deposit to join the game, and his/her deposit will be confiscated if he/she behaves dishonestly. In short, we solve the prisoner's dilemma in rational secret sharing by using blockchain and a smart contract.

One future direction would be to extend our scheme to rational delegation of computation such as [27], which may lead to more complex utility functions and incentive mechanisms, and we can analyze the players's capacity limitation of attack and defense. Another future direction would be to construct a composable scheme. For example, in a repeated game of rational secret sharing, rational players will behave honesty in each round (except the last round) in order to obtain longer cooperation and higher utility. At this time, our scheme in this paper can be combined and used to solve the problems faced in the last round. Besides, we hope that sequential equilibrium can be introduced into more traditional cryptographic primitives where the participants are either honest or malicious.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61662009, 61772008), Guizhou Provincial Department of Education Science and Technology Top Talent Support Project (Grant No. [2016]060), Science and Technology Major Support Program of Guizhou Province (Grant No. 20183001), Science and Technology Program of Guizhou Province (Grant No. [2017]5788), Ministry of Education-China Mobile Research Fund Project (Grant No. MCM20170401), Guizhou University Cultivation Project (Grant No. [2017]5788), Key Program of the National Natural Science Union Foundation of China (Grant No. U1836205), and Science and Technology Program of Guizhou Province (Grant No. [2019]1098).

References

1 Blakley G R. Safeguarding cryptographic keys. In: Proceedings of Americian Federation of Information Processing Societies (AFIPS'79) National Computer Conference, 1979. 313–317

2 Shamir A. How to share a secret. Commun ACM, 1979, 22: 612–613

3 Halpern J, Teague V. Rational secret sharing and multiparty computation: extended abstract. In: Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, 2004. 623–632

4 Dodis Y, Rabin T. Cryptography and game theory. In: Algorithmic Game Theory. Cambridge: Cambridge University Press, 2007. 181–207

5 Gordon S D, Katz J. Rational secret sharing, revisited. In: Proceedings of the 5th International Conference on Security and Cryptography for Networks, Maiori, 2006. 229–241

6 Kol G, Naor M. Games for exchanging information. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, 2008. 423–432

7 Fuchsbauer G, Katz J, Naccache D. Efficient rational secret sharing in standard communication networks. In: Proceedings of the 7th Theory of Cryptography Conference, Zurich, 2010. 419–436

8 Fudenberg D, Tirole J. Game Theory. Cambridge: MIT Press, 1991

9 Maleka S, Shareef A, Rangan C P. Rational secret sharing with repeated games. In: Proceedings of the 4th Information Security Practice and Experience Conference, Sydney, 2008. 334–346

10 Ong S J, Parkes D C, Rosen A, et al. Fairness with an honest minority and a rational majority. In: Proceedings of the 6th Theory of Cryptography Conference, San Francisco, 2009. 36–53

11 Zhang Z, Liu M. Unconditionally secure rational secret sharing in standard communication networks. In: Proceeding of the 13th International Conference on Information Security and Cryptology, Seoul, 2010. 355–369

12 Zhang Z F, Liu M L. Rational secret sharing as extensive games. Sci China Inf Sci, 2013, 56: 032107

13 Tian Y, Ma J, Peng C, et al. A rational framework for secure communication. Inf Sci, 2013, 250: 215–226

14 Tian Y L, Peng C G, Lin D D, et al. Bayesian mechanism for rational secret sharing scheme. Sci China Inf Sci, 2015, 58: 052109

15 Jin J, Zhou X, Ma C, et al. A rational secret sharing relying on reputation. In: Proceeding of the 8th International Conference on Intelligent Networking and Collaborative Systems, Ostrawva, 2016. 384–387

16 Nisan N, Ronen A. Algorithmic mechanism design. Games Economic Behav, 2001, 35: 166–196

17 Liu H, Li X H, Ma J F, et al. Reconstruction methodology for rational secret sharing based on mechanism design. Sci China Inf Sci, 2017, 60: 088101

18 Nakamoto S. Bitcoin: a peer-to-peer electronic cash system. 2008. https://bitcoin.org/en/bitcoin-paper

19 Zhou L, Wang L, Sun Y. Mistore: a blockchain-based medical insurance storage system. J Med Syst, 2018, 42: 149

20 Bartolucci S, Bernat P, Joseph D. SHARVOT: secret SHARe-based VOTing on the blockchain. In: Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain, Gothenburg, 2018. 30–34

21 Kim Y, Raman R K, Kim Y S, et al. Efficient local secret sharing for distributed blockchain systems. IEEE Commun Lett, 2019, 23: 282–285

22 Xiong F, Xiao R, Ren W, et al. A key protection scheme based on secret sharing for blockchain-based construction supply chain system. IEEE Access, 2019, 7: 126773–126786

23 Dong C, Wang Y, Aldweesh A, et al. Betrayal, distrust, and rationality: smart counter-collusion contracts for verifiable cloud computing. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, 2017. 211–227

24 Katz J. Bridging game theory and cryptography: recent results and future directions. In: Proceedings of the 5th Theory of Cryptography Conference, New York, 2008. 251–272

25 Szabo N. Formalizing and securing relationships on public networks. First Monday, 1997, 2: 1–21

26 Pedersen T P. Non-interactive and information-theoretic secure verifiable secret sharing. In: Proceedings of the 11st Annual International Cryptology Conference, Santa Barbara, 1991. 129–140

27 Tian Y, Guo J, Wu Y, et al. Towards attack and defense views of rational delegation of computation. IEEE Access, 2019, 7: 44037–44049