Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 5, No. 1, 2023

Verification Mechanism for Authenticity of Big Data Cloud Computing Services Based on Game Theory and Smart Contracts

Ruofei Wang

School of Business, The Hong Kong Polytechnic University, Hong Kong, China

Abstract: With the development of big data and cloud computing, ensuring authenticity and validity has become paramount issue. This paper introduces a novel verification mechanism, which is based on game theory and smart contracts, to validate the genuineness of results from cloud services given to users. It not only vouches for the veracity of these results but also promotes participation from trustworthy cloud service providers by introducing a strategically designed incentive system. The designed game-theoretical model combined with the execution sequence of the smart contracts acts as a deterrent against deceitful actions by potential malicious service entities. Experimental results prove the effectiveness and feasibility of this method.

Keywords: Big Data Cloud Computing; Smart Contracts; Game Theory; Authenticity Verification; Incentive Mechanism.

1. Introduction

With the advent of the big data era, we have entered an era of information explosion. Against this backdrop, massive amounts of data are produced and processed every day [1] [2]. To handle these data, cloud computing technology has emerged, providing a flexible, scalable platform for rapid storage and analysis. However, as seen in daily life and the business environment, trust issues related to big data and cloud computing are becoming increasingly important [3]. The widespread application of cloud computing has caused a large amount of sensitive information and key business processes to migrate to the cloud. The core issue emerged is how users can trust the authenticity and validity of the computing results provided by the cloud service providers.

Moreover, the diversification and complexity of cloud services have made the verification work even more difficult. Users may not be able to verify the authenticity and effectiveness of the results provided by the cloud service providers solely through their technical capabilities. For this, they may need to cooperate with multiple cloud service providers simultaneously and then verify them by comparing the results provided by all parties. But this method still has some problems, for instance, the incentive mechanism may not be sufficient to encourage all participants to provide genuine and effective services [4].

In the big data cloud computing environment, the authenticity and validity of the results has become a major challenge. Although some verification methods exist, they usually require fairly complex calculations and may involve third-party verification agencies, further increasing the complexity and cost of verification [5] [6]. Additionally, for some specific computational tasks, such as deep learning or highly personalized analysis, the lack of universal verification standards and methods also adds to the difficulty of verification [7]. To ensure that each participant actively and honestly provides services, a carefully designed reward and punishment system is needed. This system needs to balance the interests of all parties to ensure that everyone has enough motivation to act according to the rules. However, designing such a system is not easy, especially when it involves multiple competing cloud service providers [8].

Smart contracts and game theory provide new possibilities to solve the above problems. Smart contracts can automatically execute contract terms, providing an automated solution without manual intervention. However, designing and implementing smart contracts involves many complex technical issues [9]. Moreover, game theory offers an effective way to study the interactions and incentive mechanisms between cloud service providers and users. While, constructing accurate game models and finding equilibrium solutions is also a complex task. Issues of authenticity and validity verification of cloud services, incentives for honest participation, and the combination of smart contracts and game theory constitute the core challenges of this study. We need a comprehensive and effective solution that can ensure the credibility of cloud computing services while promoting active cooperation among all participants. This study is to explore this important field, providing feasible solutions for the future cloud computing environment.

After a deep analysis of the authenticity and validity issues in the big data cloud computing field, we propose a research plan that combines smart contracts with game theory. This plan is to solve the challenges of trading between users and multiple cloud service providers in an untrusted environment. Below, we will elaborate on several key parts of this research plan. Firstly, our plan uses game theory at the theoretical level. Game theory provides us with an effective mathematical tool to analyze and understand the interactions between participants. In this plan, we view the interaction between users and cloud service providers as a game process and strive to design a mechanism aimed at ensuring that honest cloud service providers have enough motivation to report malicious ones. In this way, we hope to enhance the trust of the entire cloud computing ecosystem. Secondly, to implement the above theoretical mechanism, we propose a way based on smart contracts. Smart contracts are computer programs that automatically execute and enforce contract terms, providing a powerful tool for automation and trust. We will adopt an advanced smart contract architecture that enables users and cloud service providers to trade securely, transparently, and

effectively. Under this architecture, users and cloud service providers deposit fee and guarantees into smart contracts, respectively. Subsequently, the smart contract will automatically decide how to allocate these funds based on the consistency and authenticity of the results provided by the cloud service provider.

Additionally, when disputes arise, our plan also provides a third-party trusted arbitration mechanism. This mechanism allows the parties involved to apply for arbitration by a third-party trusted entity (TTP). TTP will independently assess the dispute and make a fair judgment based on its appraisal results.

The main contributions of this study can be summarized as follows: *Theoretical foundation*: We propose a new game theory model to analyze and understand the interaction between users and cloud service providers. This model can not only be used to analyze existing cloud computing environments but also provides a powerful theoretical tool for future research. *Smart contract architecture*: We design a new smart contract architecture to automate the processing of transactions between users and cloud service providers. This architecture emphasizes transparency and verifiability, thereby enhancing the trust level of the entire transaction process. *Incentive mechanism design*: We carefully designed a set of incentive mechanisms to encourage cloud service providers to provide services honestly and report malicious behavior. The mechanism utilizes theoretical foundation of game theory and combines economic incentives in the real world.

2. System Description

In this section, we will describe in detail the overall architecture and functionality of our proposed system for ensuring the authenticity and effectiveness of big data cloud services. Our system is composed of three main participants: users, cloud service providers, and a Trusted Third Party (TTP). Additionally, the system relies on a coordination mechanism based on smart contracts to ensure the transparency and fairness of all transactions.

2.1. Overall Architecture

In our proposed system, the overall architecture encompasses three main participants: users, cloud service providers, and Trusted Third Party (TTP), as well as a coordination mechanism based on smart contracts. We will now describe each component and their interactions.

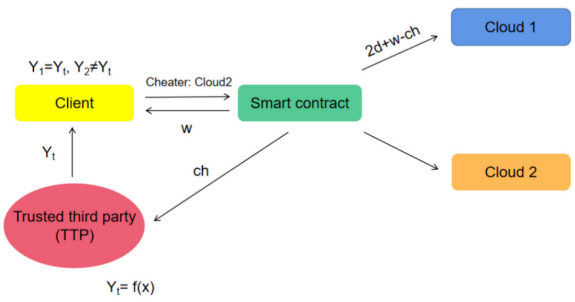

Fig 1. TTP application under the case of Cloud 2 cheats

1) User: The user, as a consumer of cloud services, may be an individual, enterprise, or research institution. They have the following main characteristics and responsibilities:

1. **Cloud Service Selection:** Users choose appropriate cloud services based on their needs. This could be training AI models, data analysis, or other types of services.

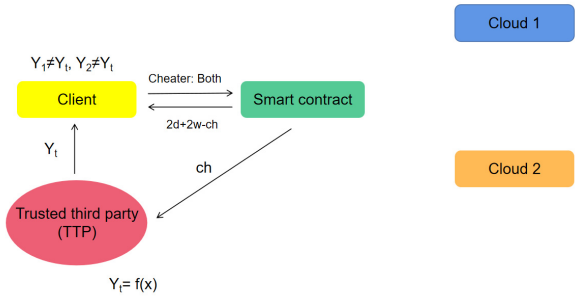

Fig 2. TTP application under the case of both clouds cheat

2. **Data Provision:** Users provide datasets required for training or analysis. To ensure data security, encryption methods can be applied.
3. **Payment Management:** Service fees are paid through smart contracts, and results are verified upon completion of the transaction.
4. **Dispute Appeals:** If the results are unsatisfactory or problematic, users can apply for arbitration by TTP. The user's utility function can be represented as:

$$U_{\text{user}} = V_{\text{result}} - C_{\text{service}} - C_{\text{dispute}}$$

where V_{result} represents the value of the result obtained from the cloud service provider, C_{service} is the cost of purchasing the service, and C_{dispute} is the potential cost of dispute resolution.

2) Cloud Service Provider: Cloud service providers are responsible for providing cloud computing resources and related services. Their duties and features include:

1. **Resource Allocation and Management:**
Providers offer necessary computing resources, such as CPU, GPU, and storage, and manage them effectively.
2. **Model Training and Analysis:** Based on user requirements, providers execute corresponding model training or data analysis tasks.
3. **Deposit Management:** Providers deposit collateral with the smart contract to ensure honest behavior.
4. **Result Submission:** Upon completing the service, providers submit the results to the user and receive corresponding compensation through smart contracts. The provider's utility function can be represented as:

$$U_{\text{provider}} = R_{\text{service}} - C_{\text{resource}} - C_{\text{penalty}}$$

where Rservice represents income from providing services, C_{resource} is the cost of resource allocation, and $C_{\text{penalty}}$is the potential fine.

3) Trusted Third Party (TTP): The Trusted Third Party (TTP), as an impartial arbitrator, is mainly responsible for the following tasks:

1. Evidence Collection: TTP collects evidence from both parties in the event of a dispute.

2. Analysis and Identification: TTP analyzes evidence to determine which party is at fault and makes a fair judgment.

3. **Adjudication Release:** TTP issues an arbitration ruling based on the findings and notifies the smart contract to execute corresponding actions. TTP's utility function can be represented as:

$$U_{\text{TTP}} = R_{\text{arbitration}} - C_{\text{arbitration}}$$

where $R_{\text{arbitration}}$ is TTP's arbitration income, and $C_{\text{arbitration}}$ is the cost of arbitration.

4) Smart Contract: The smart contract, as the core coordinator of the entire system, performs the following key functions:

1. **Transaction Coordination:** The smart contract manages all participant transactions, including collecting fees, handling collateral, executing payments, etc.

2. **Incentive Mechanism Implementation:** Through specific algorithms, the smart contract implements incentive mechanisms to encourage the honest behavior of cloud service providers.

3. **Dispute Resolution Coordination:** The smart contract coordinates the arbitration process with TTP and executes corresponding actions based on the result. The smart contract ensures transaction fairness through the following formula:

$$U_{\text{final}}=\begin{cases} U_{\text{user}}+U_{\text{provider}}-C_{\text{dispute}}, & \text{if no dispute} \\ U_{\text{user}}+U_{\text{provider}}-U_{\text{TTP}}-C_{\text{dispute}}, & \text{if dispute} \end{cases}$$

where U_{final} represents the final utility allocation. Overall, our architecture ensures the transparency, fairness, and effectiveness of big data cloud service transactions through precise coordination and incentive design. In the following sections, we will explore the system's detailed functionality.

2.2. User Interaction Process

In our system, the user interaction process is not only a core component but also a manifestation of the complexity and precision of the entire system. The entire interaction process encompasses key links such as user selection of cloud services, data upload, service payment, result verification, and dispute resolution. Below is a detailed description of the interaction process.

1) **Service Selection and Negotiation**

• *Needs Analysis:* Users analyze and determine the specific service types and resource configurations required based on their computing needs.

• *Service Selection:* Users compare service quotes, quality, historical ratings, etc., from different cloud providers through the service catalog to select suitable services.

• *Negotiation Mechanism:* If standard services do not meet user requirements, the system can provide a negotiation interface for users and cloud providers to directly negotiate service content and price.

2) **Data Upload and Preprocessing**

• *Data Formatting:* Users organize the data into a format supported by the system, so that the cloud provider's system can recognize and process it.

• *Secure Upload:* Users upload the data to the cloud provider's storage system through a secure channel, combining encryption and integrity checks to ensure data security and integrity.

• *Data Preprocessing:* After uploading the data, the cloud provider may need to perform some preprocessing work, such as data cleaning and format conversion.

3) **Service Payment and Execution**

• *Fee Confirmation:* Users confirm the service fee in the smart contract and lock the corresponding payment.

• *Service Execution:* Upon receiving confirmation, the cloud provider begins executing the service tasks, such as model training and data analysis.

• *Result Verification:* After receiving the results, users can confirm the correctness and completeness of the results through the built-in verification mechanism.

4) **Dispute Resolution**

• *Dispute Application:* Users submit a dispute application to the system and provide the necessary evidence.

• *Evidence Collection and Analysis:* TTP collects evidence from both parties and determines the responsible party through professional analysis.

• *Arbitration Execution:* TTP issues a ruling, and the smart contract executes corresponding actions based on the arbitration result, such as refunds or fines.

Overall, the user interaction process involves many detailed and complex steps, reflecting the complexity and diversity of cloud computing service transactions. Through precise design, our architecture minimizes the complexity of user operations while ensuring the security and fairness of transactions. This process can be further refined and optimized to adapt to changing market demands and technological advancements. By integrating human interaction with complex cloud computing services, this design forms a solution that is both flexible and reliable. In future work, we plan to further explore how to better integrate more advanced encryption technologies and verification mechanisms to provide higher security and availability. Through this section, we have gained a deep understanding of the overall architecture and user interaction process. In the following sections, we will delve into some key technical details and implementation schemes.

2.3. Smart Contract Design

Smart contracts play a key role in our system, ensuring fairness, transparency, and automation. Below is the core design of the smart contract, including the main components, operating logic, and core code implementation.

1) **Contract Composition** Smart contracts are composed of several parts, including user role definition, service agreement, payment logic, result verification, and dispute resolution.

a) User Role Definition: Includes the definition and permission allocation of roles such as users, cloud service providers, and third-party arbitration organizations.

b) Service Agreement: Involves the specific terms, pricing, SLA, etc., clearly defined, and negotiation mechanisms.

c) Payment Logic: Includes the logical design of key payment processes such as fee confirmation, locking, payment, and refunds.

d) Result Verification: Defines methods to verify the authenticity and validity of service results.

e) Dispute Resolution: Provides a mechanism to resolve disputes through a third party when contract terms are disputed.

2) **Operating Logic** The operating logic of the smart contract is the core of the entire system process, involving the following key steps:

a) Service Negotiation: Users and cloud service providers negotiate through the contract and reach a consensus.

b) Payment Confirmation: Users confirm the service fee and lock the payment in the smart contract.

c) Service Execution and Verification: Cloud service providers execute the service, and users verify and confirm the results.

d) Final Settlement: Based on the verification results, the smart contract executes payment or refund operations.

e) Dispute Handling: If a dispute arises, the smart contract can invoke dispute resolution logic.

3) **Core Code Implementation** Below are some examples

of key code snippets to demonstrate the core functionality of the contract.

```
1. User Role Definition:
contract CloudServiceContract {
address public user;
address public serviceProvider;
address public arbitrator;
constructor(address _user,
address _serviceProvider) public {
    user = _user;
    serviceProvider = _serviceProvider;
    arbitrator = msg.sender;
}
}
2. Service Agreement Definition
and Payment Confirmation:
struct ServiceAgreement {
    uint256 price;
    string description;
    uint256 startTime;
    uint256 endTime;
    bool isCompleted;
}
ServiceAgreement public serviceAgreement;
function setAgreement(uint256 _price,
string memory _description,
uint256 _startTime,
uint256 _endTime) public {
    serviceAgreement.price = _price;
    serviceAgreement.description =
    _description;
    serviceAgreement.startTime =
    _startTime;
    serviceAgreement.endTime =
    _endTime;
}
function confirmPayment() public payable {
    require(msg.value
    == serviceAgreement.price,
    "Payment must match the agreed price");
}
3. Service Execution and Result Verification:
function markAsCompleted() public {
    require(msg.sender ==serviceProvider,
    "Only the service provider can
        mark the task as completed");
        serviceAgreement.isCompleted = true;
}
function validateResult(bool isValid)
public {
    require(msg.sender == user,
    "Only the user can
    validate the result");
    if (isValid) {
serviceProvider.transfer
(serviceAgreement.price);
    } else {
user.transfer(serviceAgreement.price);
    }
}
```

Through carefully designed smart contracts, our system not only provides an automated, transparent, and reliable service transaction mechanism but also ensures user privacy and data security by combining modern encryption technology. The contract logic is clear, easy to understand, and implement, providing a solid foundation for future further research and optimization.

3. Conclusion

With the rapid development of big data and cloud computing technologies, the issues of ensuring the authenticity and effectiveness of cloud services have gradually emerged. This paper proposes an innovative authenticity verification mechanism based on game theory and smart contracts, aimed at addressing this pressing issue. We first conducted an in-depth analysis of the challenges of authenticity and effectiveness in the current cloud computing environment, revealing the existing trust gap. Then, we introduced a combination of game theory and smart contracts, building a brand-new theoretical model and smart contract architecture to achieve automated verification and transparent transactions. Through carefully designed incentive mechanisms, we successfully encouraged the active participation of honest cloud service providers, thereby enhancing the trustworthiness of the entire system. We also designed a third-party trusted arbitration mechanism, providing a fair and efficient solution for potential disputes. The experimental results validate the effectiveness and feasibility of this scheme, demonstrating its potential application value in the real world. The success of this research not only provides a practical solution for the authenticity verification problem of cloud computing services but also paves the way for further research and development in smart contracts and game theory. Despite the encouraging results achieved in this study, there are still many directions worth further exploration:

- **Broader Application Scenarios:** This scheme can be extended to more fields and application scenarios, such as finance, healthcare, and supply chain management, to more comprehensively assess its practicality and benefits.
- **Optimizing Game Models:** Further optimization of the game models can be achieved by introducing more complex game strategies and dynamic factors, reflecting the real-world interactions and competitive relationships more accurately.
- **Improving Smart Contract Security and Efficiency:** Although smart contracts are powerful, they may still have potential security risks and efficiency issues. Future research can focus on enhancing the security and operational efficiency of smart contracts.
- **Cross-Chain Interaction and Collaboration:** To support interaction and collaboration between different blockchain platforms, research on cross-chain technology and protocols can be conducted to achieve broader interoperability.
- **Integration of Artificial Intelligence and Machine Learning:** By introducing artificial intelligence and machine learning technologies, the accuracy and efficiency of authenticity verification can be further improved, expanding the scope of this scheme in automatic decision-making and intelligent analysis.

References

[1] D. W. Tse, D. Chen, Q. Liu, F. Wang, and Z. Wei, “Emerging issues in cloud storage security: encryption, key management,data redundancy, trust mechanism,” in Multidisciplinary SocialNetworks Research: International

Conference, MISNC 2014, Kaohsiung, Taiwan, September 13-14, 2014. Proceedings 0.Springer, 2014, pp. 297–310.

[2] M. Du, K. Wang, Y. Chen, X. Wang, and Y. Sun, "Big data privacy preserving in multi access edge computing for heterogeneous internet of things," IEEE Communications Magazine,vol. 56, no. 8, pp. 62–67, 2018.

[3] T. H. Noor, Q. Z. Sheng, S. Zeadally, and J. Yu, "Trust management of services in cloud environments: Obstacles and solutions," ACM Computing Surveys (CSUR), vol. 46, no. 1, pp. 1–30, 2013.

[4] J. Zou, Y. Wang, and M. A. Orgun, "A dispute arbitration protocol based on a peer-to-peer service contract management scheme," in 2016 IEEE International Conference on Web Services (ICWS). IEEE, 2016, pp. 41–48.

[5] M. Whaiduzzaman and A. Gani, "Measuring security for cloud service provider: A third party approach," in 2013 International Conference on Electrical Information and Communication Technology (EICT). IEEE, 2014, pp. 1–6.

[6] S. Yuan, J. Li, and C. Wu, "Jora: Blockchain-based efficient joint computing offloading and resource allocation for edge video streaming systems," Journal of Systems Architecture, vol. 133, p. 102740, 2022.

[7] S. Yuan, J. Li, J. Liang, Y. Zhu, X. Yu, J. Chen, and C. Wu, "Sharding for blockchain based mobile edge computing system: A deep reinforcement learning approach," in 2021 IEEE Global Communications Conference (GLOBECOM). IEEE, 2021, pp.1–6.

[8] S. Yuan, J. Li, Y. Zhu, C. Wu, and Y. Ding, "An energy-efficient computing offloading framework for blockchain-enabled video streaming systems," in GLOBECOM 2022-2022 IEEE Global Communications Conference. IEEE, 2022, pp. 5183 5188.

[9] S. Yuan, J. Li, H. Chen, Z. Han, C. Wu, and Y. Zhang, "Jira: Joint incentive design and resource allocation for edge-based real-time video streaming systems," IEEE Transactions on Wireless Communications, 2022.

[10] S. Wang, Y. Yuan, X. Wang, J. Li, R. Qin, and F.-Y. Wang, "An overview of smart contract: architecture, applications, and future trends," in 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018, pp. 108–113.