Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3704720, 12 pages
https://doi.org/10.1155/2022/3704720

Research Article

Trusted Cloud Service System Based on Block Chain Technology

Tilei Gao, Xiaohui Jia, Rong Jiang, Yuanyuan He, Tao Zhang, and Ming Yang

School of Information, Yunnan University of Finance and Economics, Kunming 650221, China

Correspondence should be addressed to Ming Yang; yangming@ynufe.edu.cn

Received 17 May 2022; Revised 5 August 2022; Accepted 9 August 2022; Published 21 August 2022

Academic Editor: Yin Zhang

Copyright © 2022 Tilei Gao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development and popularization of cloud computing technology, more and more users choose cloud services to build their application systems. With the improvement of users' understanding of software systems, in addition to functionality, trustworthiness has become another key issue concerned by users. Based on the existing research on cloud service trustworthiness measurement and evaluation, this paper proposes a cloud service-trusted delivery model based on asymmetric encryption and hash function and a trusted runtime model of cloud service system based on block chain technology. The proposed models will effectively solve the untrusted problems such as denial and tampering in the process of cloud service acquisition, as well as the system anomaly at runtime. Finally, through targeted experiments to verify the effectiveness and feasibility of the proposed models, and through experimental analysis, this paper expounds on the principle and mechanism of model operation and trustworthiness guarantee.

1. Introduction

The popularity of the Internet has had a significant impact on the development of computing mode and has realized a significant change from providing scientific computing power to providing network services [1]. At present, cloud computing has gradually become the main computing mode. According to Flexera's 2020 cloud status report, 59% of enterprises expect cloud usage to exceed previous plans [2]. It can be seen that the demand for cloud services in the global market is gradually increasing, and more and more institutions begin to choose to use cloud services to expand their applications. At the same time, the introduction of new technologies such as software-defined networks, artificial intelligence, and big data also poses a greater challenge to the trustworthiness of cloud computing [3]. With the increase of cloud services and the gradual deepening of users' understanding of software systems, users' needs are no longer limited to the satisfiability of functions but pay more and more attention to the trustworthiness of software systems.

In the cloud computing environment, the trustworthiness of cloud service systems is usually considered from the following three aspects: the selection of trusted cloud services, the trustworthiness of cloud services acquisition or delivery process, and the trustworthiness of cloud service running time. Through the review of the existing research on the trustworthiness of cloud services, it can be seen that most of the existing research results focus on the trustworthiness measurement, evaluation, and selection of cloud services. These results are indeed quite helpful for users to select reliable and appropriate services and build systems. For the cloud service providing platform, it is not difficult to find the services required by users from different cloud service marts. However, due to the different trustworthiness of different service providers in different fairs and the existence of unsafe factors in the network transmission process, how to ensure the trustworthiness of the transmission process of cloud services has become one of the focus issues of cloud platforms. In addition, during the operation of the cloud service system, how to provide continuous services to users without stopping the system is also one of the key concerns of the cloud platform.

The main research content of this paper focuses on the trustworthiness of the cloud service delivery process and the trustworthiness of the system runtime process. In the research of the trustworthiness of cloud service delivery process, a trusted delivery process model based on the hash function and asymmetric encryption technology is

proposed. In the research of runtime trustworthiness, a runtime trustworthiness model based on the block chain technology is proposed. Through the design and analysis of simulation experiments, the feasibility and applicability of the models are verified.

The structure of the article is as follows:

Section 1 covers the introduction of this research study

Section 2 introduces the research status and achievements in related fields

Section 3 gives the definition and explanation of relevant basic concepts

Section 4 presents the trusted cloud service delivery process model

Section 5 presents the trusted cloud service system runtime model

Section 6 verifies the feasibility of the model through experimental analysis and expounds on the advantages and disadvantages of the models proposed

Section 7 is the conclusion and future works part

2. Literature Review

2.1. Research Status of Trustworthiness. Trustworthiness in the field of information science is often related to system requirements. In requirements engineering, nonfunctional requirements are usually regarded as the quality attribute of software. The quality of the software is actually an objective evaluation of software system, which will not change due to the difference in environment, personnel, and conditions [4]. Trustworthiness is an attribute with strong subjective preference. Its trustworthiness will show great differences in different application scenarios [5]. This difference is mainly reflected in the attention of users to the subattributes contained in software trustworthiness: some users pay attention to whether the functional requirements can be well realized, some users pay attention to whether the efficiency is high enough, and some users pay attention to whether the reliability is better guaranteed. These are the categories of software trustworthiness.

In terms of the trustworthiness content of software service systems, since the concept of trusted computing was put forward, the research on software trustworthiness [6] and trusted software [7] has gradually become one of the research hotspots in the field of software engineering. Its main research content is the construction and application of the trustworthiness model. According to the definition of TCG, an entity is credible if it always develops towards the expected goal. Shen [8] pointed out that trustworthiness includes reliability and safety. Yang et al. [9] pointed out that trustworthiness includes ability trustworthiness, integrity trustworthiness, predictability, correctness, privacy, and loss cost. In view of the trustworthiness of cloud platforms, Zhao et al. [10] pointed out that the content and analysis evaluation basis of cloud platform trustworthiness is not perfect and there is a lack of analysis and evaluation at the theoretical level. As a kind of evaluation, the trustworthiness evaluation of a cloud platform is inevitably vulnerable to human subjective factors in the evaluation process [11].

Research achievements in trustworthiness measurement and evaluation mainly include demand-driven software trustworthiness evaluation and evolution model [12], software trustworthiness evaluation model based on evidence theory [13], runtime software trustworthiness evidence collection mechanism based on TPM [14], software service trustworthiness evaluation method based on subjective and objective comprehensive weighting [15], evaluation method based on fuzzy theory [16], evaluation method based on D-S evidence theory [17], evaluation method based on risk matrix [18], trusted computing method based on trusted chain [19], and prediction evaluation method based on Bayesian network [20]. At the same time, there are also some general measurement and evaluation methods that can be used in various fields, such as the AHP method and extension method based on AHP [21], analysis method based on the decision tree and its deformation fault tree [22], and method based on information entropy and entropy weight theory [23].

Although there are many existing methods for system trustworthiness and trustworthiness measurement and evaluation, they are all for the measurement and evaluation of the service itself and are all preparations for the selection of services. In the traditional system field, users communicate directly with developers or system providers and need a set of methods to judge the trustworthiness of the system, so as to facilitate users to verify and accept the purchased system. However, under the new computing mode with cloud computing as the main technology, both users and cloud service system construction platforms obtain software services from unknown places to build the system. In this new computing environment, in addition to the traditional measurement and evaluation methods of software services and the software systems themselves, in order to ensure the trustworthiness of the remote systems, the trustworthiness of the service delivery process and service operation process should also be concerned.

2.2. Research Status of Block Chain. Block chain technology originated from bitcoin [24] and was originally designed to solve the problem of overreliance on trusted third parties in electronic payment. Block chain reorganizes mature technologies such as hash function, Merkle tree, and proof of work (POW) [25] and combines cryptography technologies such as public key encryption, digital signature, and zero-knowledge proof to become a new distributed infrastructure and computing paradigm [26]. Block chain solves the problem of consistency in distributed networks and subverts the traditional technical architecture of relying on trusted third parties to realize large-scale organization management and control. Its application has gradually extended to many fields, such as finance [27, 28], digital economy [29, 30], Internet of Things [31, 32], intelligent manufacturing [33, 34], and data security [35, 36], and has become a research hotspot in the global academic community.

Block chain covers a variety of technologies, the related concepts are easy to confuse, and there are many application scenarios. Therefore, relevant reviews have been made to sort out the latest progress, technical differences, and

connections of block chain and summarize the technical form and application value from the perspective of technical architecture, technical challenges, and application scenarios. Yuan and Wang [26] gave the basic model of block chain. Taking bitcoin as an example, the unlicensed chain is divided into data layer, network layer, consensus layer, incentive layer, contract layer, and application layer; Shao et al. [37] compared the technical characteristics of various enterprise block chains (license chains) in combination with the details of open source projects; Yang et al. [38] summarized the characteristics, challenges, and development trends of block chain-based network service architecture; Han et al. [39] systematically summarized the research status of block chain security; Ali et al. [40] summarized the application research progress and trend of block chain in the Internet of Things.

Due to the two characteristics of process trustworthiness and decentralization, block chain can build a trusted foundation in a low-cost way in the scenario of multistakeholder participation, aiming to reshape the social credit system. In the cloud computing environment, the sources of cloud services that constitute service systems are diverse, and the existing mechanism is difficult to ensure the trustworthiness of service sources. In addition, the trustworthiness of the operation process of a cloud service system is the basis to ensure the normal operation of user's business. In the cloud environment, systems are far away from users. How to monitor the trustworthiness of the system in real time is also one of the key issues concerned by users and cloud platforms. The use of asymmetric encryption and hash function in block chain technology provides an idea for us to design a trusted service delivery process model. The characteristics of distributed ledger and centralization in the cloud in block chain provide a basis for the trustworthiness of users' real-time monitoring system. Therefore, based on block chain technology, this paper will build the trusted delivery model of cloud services and the runtime trusted verification model of cloud service systems.

3. Concepts

The main research object of this paper is the trustworthiness verification process of cloud service systems. The trustworthiness verification process includes two aspects: the trustworthiness verification of the transmission process and the trustworthiness verification of the running time.

The delivery process of cloud services, that is, the acquisition process of cloud services, refers to the process in which users select cloud services suitable for their application scenarios according to their needs and provide the selection results to the cloud platform and the cloud platform obtains cloud services from the service provider. This process needs to address the following trustworthiness issues:

(1) The confidentiality of the transmission process to avoid unauthorized access or interception

(2) The denial of the sender avoids the phenomenon that the sender sends malicious code and does not admit it

(3) The denial of the receiving party to avoid the phenomenon that the receiving party receives the service without acknowledging it

The main dependent technologies include asymmetric encryption technology in cryptography and hash function.

In terms of trustworthiness verification of cloud service systems at runtime, in the cloud computing environment, most cloud service individuals are function blocks that can run independently. After the cloud platform obtains the cloud services specified by the user, it is gradually integrated into its platform. In the integration process, a status is recorded for each service integrated. Taking the integrated state as the standard, if the state is inconsistent during operation, it indicates that the trustworthiness of the system has been damaged. In this way, the trustworthiness of the system at different times can be detected.

Starting from the first service construction, the status after each service addition, deletion, and modification is regarded as a block. Gradually build a block chain with each state in the construction process of cloud service system as the main body. Finally, the trustworthiness verification model of the operation process of cloud service system is constructed.

In order to realize the above two verification models, it is necessary to formally describe the relevant concepts. The following content of this section will formally describe the concepts used in the modeling process:

Definition 1 (cloud service system). Cloud service system (CSS) is a binary, $\mathrm{CSS} = (S, Q)$, during which

(1) S is the collection of cloud services which make up the cloud service system, and $S = \{\mathrm{ser}_1, \mathrm{ser}_2, \cdots, \mathrm{ser}_n\}$

(2) Q is the relationship matrix of CSS

Definition 2 (cloud service). Let ser be cloud service. Ser is a 5-tuples, and $\mathrm{ser} = (\mathrm{id}, \mathrm{name}, \mathrm{source}, \mathrm{description}, \mathrm{codes})$, during which

(1) Id is the number of the services making up system. And in a system, id is unique and assigned automatically

(2) Name is the name given by the system to be built according to its content specification, not by its developer

(3) Source represents the source of a cloud service

(4) Description represents the overall description of a cloud service

(5) Codes represent the source code package of a cloud service

Definition 3 (relationship matrix). Relationship matrix $Q = (q_{11}, q_{12}, \cdots, q_{ij}, \cdots, q_{nn})$. The value of q_{ij} means whether there is a message communication from ser_i to ser_j, and

$$Q = \begin{bmatrix} q_{11} & \cdots & q_{1n} \\ \vdots & q_{ij} & \vdots \\ q_{n1} & \cdots & q_{nn} \end{bmatrix}, \tag{1}$$

$$q_{ij} = \begin{cases} 0, \text{ if } ser_i \longrightarrow ser_j = \varnothing, \\ 1, \text{ if } ser_i \longrightarrow ser_j \neq \varnothing. \end{cases} \tag{2}$$

In order to facilitate the calculation later, let R be the coding form of relation matrix Q, and $R = r_1 r_2 \cdots r_k \cdots r_{n*n}$. For instance, if the relationship matrix

$$Q = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \tag{3}$$

then the value of R is $R = 010001110$.

4. Trusted Cloud Service Delivery Process Model

The delivery process of cloud services is the process that the cloud platform obtains the required cloud services from different cloud service providers. The process includes three stages: the sending stage of cloud services, the transmission stage of cloud services, and the receiving stage of cloud services. The trusted cloud service delivery process should ensure the nonrepudiation of the sender and receiver of the service and the confidentiality and integrity of the transmission process. In order to achieve these trusted goals, this section proposes a trusted cloud service delivery process based on asymmetric encryption and hash function and designs a trusted delivery process from the three stages included in the delivery process.

The delivery process of cloud services includes service providers and service recipients. Assume that the service provider is A and the service recipient is B. The trusted cloud service delivery process must use the asymmetric encryption algorithm twice, which needs to generate the public key and private key of A and B, respectively. Suppose SK_A and PK_A represent the private and public keys of A, respectively, and SK_B and PK_B represent the private and public keys of B, respectively.

4.1. Trusted Sender. The trusted sender can be described as follows:

Definition 4 (trusted sender). Let TS be the trusted sender. TS is a 4-tuples, and $\text{TS} = (S\_S, S\_\text{ABR}, S\_\text{SIG}, S\_C)$, during which

(1) S_S is the collection of cloud services defined before

(2) S_ABR is the collection of abstracts, and each abstract is a hash of its service, that is,

$$\forall \text{abr}_i \in S\_\text{ABR} : \text{abr}_i \longleftarrow \text{Hash}(\text{ser}_i) \tag{4}$$

(3) S_SIG is the collection of signatures, and each signature is obtained by encrypting the abstract, that is,

$$\forall \text{sig}_i \in \text{S}_{\text{SIG}} : \text{sig}_i \longleftarrow E_{\text{SK}_A}(\text{arb}_i) \tag{5}$$

$E_{\text{SK}\ A}(\ \text{arb}_i)$ means encrypting arb_i with SK_A as the key. The encryption algorithm is selected according to the specific situation.

(4) S_C is the collection of cypher texts, which is the form of transmission on the network

$$\forall c_i \in S\_C : c_i \longleftarrow E_{\text{PK}_B}(\text{ser}_i + \text{sig}_i) \tag{6}$$

$E_{\text{PK}_B}\ (\text{ser}_i\ + \text{sig}\ i)$ means encrypting ser_i and sig_i as a whole, and the key is PK_B. The encryption algorithm is selected according to the specific situation. The algorithm can be the same or different with the algorithm in S_SIG.

The trusted sender model is shown in Figure 1.

4.2. Trusted Receiver. The trusted receiver can be described as follows:

Definition 5 (trusted receiver). Let TR be the trusted sender. TR is a 5-tuples, and $\text{TR} = (R\_S, R\_\text{ABR}, R\_\text{ABR}', R\_\text{SIG}, R\_C; F)$, during which

(1) R_S is the collection of cloud services defined before

(2) R_ABR is the collection of abstracts, and each abstract is a hash of its service, that is,

$$\forall \text{abr}_i \in R\_\text{ABR} : \text{abr}_i \longleftarrow \text{Hash}(\text{ser}_i) \tag{7}$$

(3) R_ABR' is the collection of abstracts, which obtained by decrypting the signature sig_i, that is,

$$\forall \text{abr}_i' \in R\_\text{ABR}' : \text{abr}_i' \longleftarrow D_{\text{PK}_A}(\text{sig}_i) \tag{8}$$

(4) R_SIG is the collection of signatures, and each signature is obtained by encrypting the abstract, that is,

$$\forall \text{sig}_i \in R\_\text{SIG} : \text{sig}_i \longleftarrow D_{\text{SK}_B}(c_i) \tag{9}$$

$D_{SK\ B}(c\ _i)$ means decrypting c_i with SK_B as the key. The decryption algorithm is the inverse of $E_{\text{PK}_B}(\text{ser}_i + \text{sig}_i)$.

(5) R_C is the collection of cypher texts, which are contents received from the transmission process

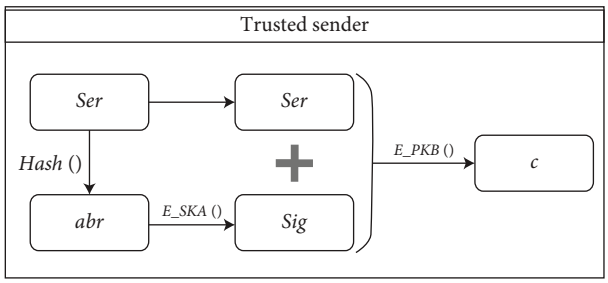

Figure 1: Trusted sender.

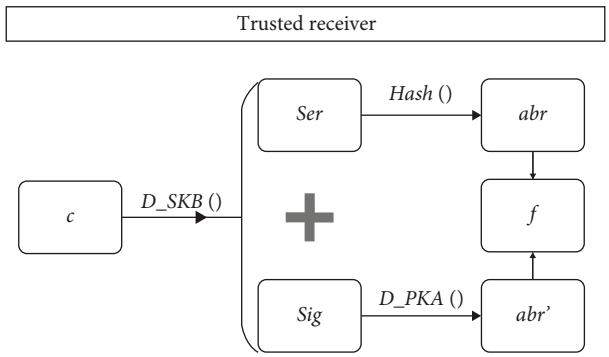

Figure 2: Trusted receiver.

(6) F is a matching function, which is used to make sure whether abr_i and $\text{abr}_i^{'}$ are the same, and

$$F : f_i = \begin{cases} 0, \text{if } \text{abr}_i = \text{abr}_i', \\ 1, \text{if } \text{abr}_i \neq \text{abr}_i' \end{cases} \tag{10}$$

The trusted receiver model is shown in Figure 2.

4.3. Trusted Receiver. Based on the models of trusted sender and trusted receiver, the whole trusted cloud service delivery process model is shown in Figure 3.

The sender obtains the digest information of the service to be sold through hash operation and generates a signature from the summary information through the sender's private key SK_A. Package the signature with the cloud service and encrypt it with the public key PK_B of the receiver to get the cypher text packet sent to the receiver. In this process, the digest is used by the receiver to check whether the service has been tampered with during network transmission; the sender's signature generated by the sender's unique private key SK_A can determine the identity of the sender of the cloud service.

After receiving the cypher text packet, the receiver can decrypt the packet by using its unique private key SK_B. If the decryption is successful, the legal identity of the receiver can be confirmed. Split the decrypted packet into service body and signature. The received service body is used to recalculate the message digest, and the signature can be decrypted through the sender's public key PK_A to obtain its original information, that is, the message digest calculated by the sender through the hash function. If the digest calculated by the service is consistent with the digest contained in the signature, it indicates that the service body has not been tampered with in the process of network transmission. Otherwise, the service is tampered with during transmission, and it is not trusted.

5. Trusted Cloud Service System Runtime Model

Most of the existing research results on the trustworthiness of cloud service systems are aimed at the trustworthiness evaluation under the state of system shutdown, that is, the static evaluation results. Once the system runs, it is difficult to obtain real-time, effective, and reliable evaluation results. Based on the previous research results, this paper takes the static evaluation results after system construction as the standard, takes the state of cloud service system at different times as the main block body, and constructs a system state chain based on block chain technology. After the system runs, by calculating the system state at different times and comparing the results with the corresponding blocks in the state chain, the system can be verified whether it has been tampered with, so as to ensure the trustworthiness of the system running time.

Definition 6 (state chain). Let state chain be SC, and SC is a 3-tuples. $\text{SC} = (\text{st}, B, N)$, during which

(1) st represents the start block of a state chain of a system, and

$$\text{st} = \text{Hash}(\text{sys}_{\text{info}}) \tag{11}$$

In the formula, sys_info represents the overall description of the system and its contents and formats are designed based on the actual needs.

(2) B is the collection of state blocks except the start block st. $B = \{b_1, b_2, \cdots\}$

(3) N is the collection of flows between each blocks

Definition 7 (state block). State block b is a binary, and $b = (\text{Header}, \text{body})$.

Definition 8 (state block body). State block body is a binary, and $\text{Body} = (S, R)$. S is the collection of cloud services in Definition 1 and R is the value of relationship matrix Q in Definition 3.

Definition 9 (state block header). State block header is a binary, and $\text{Header} = (\text{PreHash}, \text{RootHash})$, during which

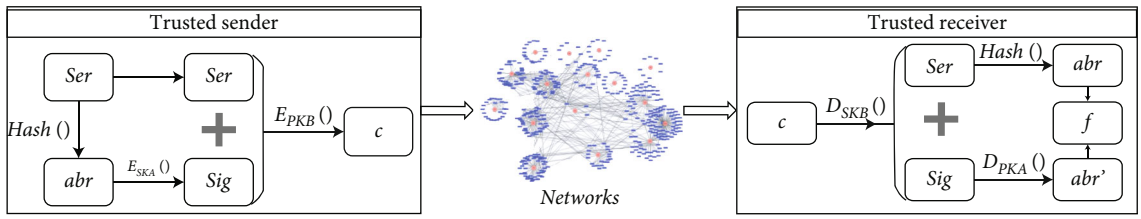

Figure 3: Trusted delivery process.

Header contains the previous block hash PreHash and current block hash RootHash.

$$\text{PreHash} = \text{Hash}(\text{Header}_{i-1}), \tag{12}$$

$$\text{RootHash} = \text{Hash}(b_i). \tag{13}$$

Based on Definitions 6, 7, 8, and 9 and the basic concepts and principles of block chain, the state chain model is shown in Figures 4 and 5.

In Figures 4 and 5, b_i is the abbreviation of body_i, representing a certain block. s_i is the abbreviation of service_i, representing a certain service. r_i is the abbreviation of relationship_i, representing a certain matrix. hashs_i is the hash value of s_i. hashs_i is the hash value of s_i. hashs_{ij} is a combination of multiple hash values from hashs_i to hashs_j.

For a new cloud service system, the relevant description of the system itself can be used as the original information of the head node, and the calculated hash value can be used as the head node. For the heritage system, the heritage information to be improved can be packaged and its hash value can be calculated as the head node. The head node is not used as the standard or basis for judging the trustworthiness of the system.

Ordinary nodes are composed of general state blocks. The state block consists of two parts: the head and the main body of the state block. The header contains the hash value of the previous state block, while the main part is composed of the current system composition. Treat each service constituting the system as a transaction record in the block chain and calculate the hash value. Then, find out the interaction information of each service in the system, obtain the structure matrix of the system, take it as the last transaction record of the state block, and calculate the hash value. Then, all hash values calculated by the service and structure are combined to finally obtain the hash value RootHash about the current state block body in the state block header. The size of each state block body varies according to the number of services. However, according to the characteristics of the hash function, no matter how much data is involved in the operation, using the same hash function will get the same output. Therefore, except for the head node, the length information of each block in the state chain is the same regardless of the state of the system.

In principle, the state of the system is unlimited, and services can be added, deleted, and modified at any time. Therefore, the state chain of the cloud service system can be unlimited in principle. However, in the actual use process,

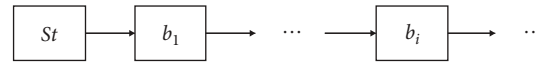

Figure 4: State chain.

during the system construction process, the state blocks will be added more frequently. After the system is stable, there will be relatively few adjustments and modifications to the system. Therefore, the growth of the state chain will slow down significantly. Any software system has its life cycle and will be abandoned. When users give up updating or building a new system, the existing state chain can be ended. The new system rebuilds the new head node and rebuilds the chain according to the corresponding rules. The status chain is a kind of alliance chain and belongs to the cloud service system platform. According to the principle of block chain distributed accounting, users can store the status chain and detect the trustworthiness of the system at the current time according to the status chain and the current running state of the system.

In particular, during the construction and operation of the service system, any active changes are system state changes, which should all be recorded in the state chain. Whether adding, deleting, or replacing services is a change in the overall state of the system. At the same time, only record one state change at a time. That is, if there are multiple services needed to be changed at the same time, treat each service as a separate individual and make changes in turn. Every service change is recorded into the chain as a change of state. The order of modification can be preset according to the actual needs of users or according to the classic principle of first come, first serve. In addition, each state block in the state chain is relatively independent. If there are multiple repeated operations, any variant will be added to the state chain as a new state according to the previous rule. For example, if service s_1 is added at a certain time and deleted at the next time, then the original service s_1 is added. Then, this process will produce three state blocks, namely, adding s_1 block, deleting s_1 block, and adding s_1 block. Due to the different time of adding, the values of these three state blocks will not be the same.

In terms of the trustworthiness verification method of a cloud service system, whether it is a newly built cloud service system or a cloud service system in stable operation, any active modification or adjustment is recorded as a state block and added to the state chain. Before any active modification operation, the new trustworthiness state of the modified system can be predicted by the existing trustworthiness

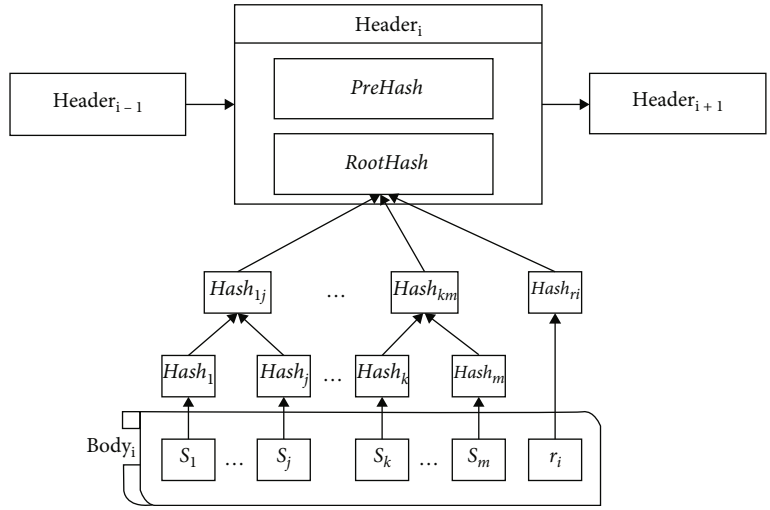

Figure 5: Block body structure of state chain.

measurement and evaluation methods. In the same environment, if the services constituting the system and the interaction relationship between services remain unchanged, its trustworthiness state remains unchanged. Then, any passive change in the system can be regarded as the destruction of trustworthiness. Therefore, the current state information of the service system can be calculated in real time and compared with the information at the initial stage of the state stored in the state chain. If it is the same, it indicates that it has not been invaded and its trustworthiness can be maintained. If it is different, it indicates that there are passively modified contents in the system, which can be traced back to the front block to determine the scope of malicious modification. Therefore, the construction of a state chain can not only determine the trustworthiness of the current system but also help the platform find the tampered services within a certain range.

6. Experiments and Analysis

The purpose of the experiment is to verify the effectiveness of the trusted model of the cloud service delivery process and the trusted model of system runtime. There is a direct correlation between the two models, but they are relatively independent. In terms of relevance, the transmitted result is the input of the system integration process. If the service received by the cloud platform is not trusted, the next process of building the system cannot be carried out. In terms of relative independence, after the cloud platform receives trusted cloud services, the subsequent system and chain-building process will not be affected by the delivery process. In general, the transfer process and the construction process of the state chain also belong to different modules. Therefore, in order to more specifically verify the effectiveness of the model and analyze the feasibility and trustworthiness of the model itself, this section designs experiments for the two models, respectively, and makes an independent analyses for each experiment. If the trustworthiness of each model is verified, the overall trustworthiness can be maintained.

6.1. Experiment and Analysis of Trusted Model of Cloud Service Delivery Process

6.1.1. Experiment Design. Suppose ser is a cloud service. According to Definition 2, ser can be described like this,

The source of the service can be a link or a user, which can be set according to the actual situation. The description of a service can generally be represented by the hash value of the service. The source code of the service or the service itself is generally a package. Here, in order to simplify the operation, the access link of the service is used as the code. The effect is the same as that of the package.

In the verification process, SHA256 and RSA have been selected as the hash function and asymmetric encryption algorithm. Suppose A is the sender and B is the receiver. Then, SK_A presents the private key of A; PK_A presents the public key of A; SK_B presents the private key of B; PK_B presents the public key of B. To simplify the operation, the key length is 512 bits and the format is PKCS#1.

Step 1. According to formula (4), the digest abr of ser can be obtained.

abr = "*6f55d6f3081f8426433d184180056c15789225cc6a7fee23c6049b1fd02bdfb1*"

Step 2. According to formula (5), the signature sig of ser can be obtained.

sig = "*k7QvgUZmLjLpjvR+p3/BQNawxLzlj7HtvwKgzFEAJ40/ACDtDSL00G6qcvx2aSVh+g1dJkPnSjz/EiABSvhmfw==*"

Step 3. According to formula (6), the cyphertext can be obtained.

```
ser = (0001,
"Gzipped source tarball",
"https://www.python.org/downloads/release/python-3104/",
"7011fa5e61dc467ac9a98c3d62cfe2be",
"https://www.python.org/ftp/python/3.10.4/Python-3.10.4.tgz")
```

Code 1

```
SK_A =
"———BEGIN RSA PRIVATE KEY———
MIIBOwIBAAJBAOvcQ2C/AAmlAcFs
jZbieobeeB2bFMfS+jsHlhezr344fr7ih
MBinbdWsUUScN9nR0aJxjF3
YKB0CaKfKSF0xMkCAwEAAQJBAJ6q9sjGtQfH8X5l
wHqYsUS5tKR2B2zGCYBcgiQ/xPdrT3sdz
WcyXSppL2uceEwimHtOegl+I6uih4VC
58UNouECIQD27HhcVpUd0i/54
NxRMDunZAyxlNdXYsv360dCr2WuGwIhAPSHs
DSOCX2vIvFt5bz4FYhkVFtn2qEVfErmhmq
acPbrAiAJ2j+nN5E1omhlqRJBbxJCS
Jy1DUJWa0vGNa4fPA5rlwIgInFGSX
DEN3bGtjjjhiVvawGuvB05tzy+gBJOVo+
gX7cCIQDlnx96LQoWPOErGn4
etaM4aJkcHBMSIR6nbVAbaMnQwA==
———END RSA PRIVATE KEY———"PK_A =
"———BEGIN RSA PUBLIC KEY———
MEgCQQDr3ENgvwAJpQHBbI2W4nqG3ngdmxTH0vo7B5YXs69+
OH6+4oTAYp23VrFFEnDfZ0dGicYxd2CgdAminykhdMTJAgMBAAE=
———END RSA PUBLIC KEY———"SK_B =
"———BEGIN RSA PRIVATE KEY———
MIIBOgIBAAJBALJseZoYxQChT8PDv0tgrbkvPx/
ye5nu71Ye5hPvlfgM4VXubos
i119ZYLX0z5FLMqbAco8/Fa3sMRyZGYlxd6cC
AwEAAQJABJ4KB5LchkemaMqICMtXs
5Mlbw43ZKRqTTA/hASPPPwNSJrE61Rhkgnl0vErX/l
YMH40nQ1glcPjGWKJ5O8cAQIh
ANlR/BizAfpJeeq0pPythTsMBz0vWFdBN4A/V
zNmYOU3AiEA0i4z6qjWdq61N7CO3sro
9h6WoFrFSgaYyW1P5QJPORECIQCW8NGnGhYCkCw
kr4l0ktTZuTYB8jNqjzqMUfIwGii
sqwIgI8MGxGOr8g+x9+LLvG7MCqyTtn
8bWIgc0REPagjlj/EC
IGtS7b7QfoGsLY6jtC0S6MC5t
UBJmgArxH0xJnG7hzPZ
———END RSA PRIVATE KEY———"
PK_B =
"———BEGIN RSA PUBLIC KEY———
MEgCQQCybHmaGMUAoU/Dw79LYK25Lz8f8nuZ7u9WHuYT75
X4DOFV7m6LItdfWWC19M+RSzKmwHKPPxWt7DEcmRmJcXenAgMBAAE=
———END RSA PUBLIC KEY———"
```

Code 2

c = "*ixnJEQEQVEzcqZ5Ny1M7ILprCA1CNslfGnBfRGa*
EPETGrMImSDjqNNZwiWT5LAZTyh3u8KoV9sdro9d1xZy
mdA==

dAulJZqQCRA8ZstTD5hbyF3dow2i2otJDg//hIqV3EXL
nSmA8gk50GHo2NuS1tv8ulIvPIDojLyN9ceBcg5BtQ==

DTqG2pgDWyWJ9Q17U9KORn4txRHqn3Pgxpy5S2d3
zdUwP3xKAcZQyMXxONijI32WfAW53CqDyR4d9xsB3W
RkJw==

bb2cN1Yy79Mjt2TAvXM2Guj5hKqfU9b2JWJN2su4h/nd
T9aN9OKxaV6ZSIDqxtYf2XN1klHTrgkQRhZ2r9UTxg=="

```
ser_1 = (0001,
"Gzipped source tarball",
"https://www.python.org/downloads/release/python-3104/",
"7011fa5e61dc467ac9a98c3d62cfe2be",
"https://www.python.org/ftp/python/3.10.4/Python-3.10.4.tgz").
ser_2 = (0002,
"XZ compressed source tarball",
"https://www.python.org/downloads/release/python-3104/",
"21f2e113e087083a1e8cf10553d93599",
"https://www.python.org/ftp/python/3.10.4/Python-3.10.4.tar.xz").
ser_3 = (0003,
"manpages-2.9.5.tar.xz",
"https:// http://edge.kernel.org/pub/software/scm/git/",
"cb6822a6eedd1682bbe815eb26d9fdbe",
"https:// http://edge.kernel.org/pub/software/scm/git/git-manpages-2.9.5.tar.xz")
ser_4 = (0004,
"B23Downloader-v0.9.5.7",
"https://github.com/vooidzero",
"4192a001fc65a0ad8016bf1328da28d0",
"https://github.com/vooidzero/B23Downloader.git")
```

Code 3

After the cyphertext c obtained, the sender's work has finished, and the cyphertext transmitted on the networks. When the receiver receives the cyphertext, the verification process begins.

Step 4. According to formula (9), cyphertext is decrypted, and the contents of ser and its signature sig can be obtained.

Step 5. According to formula (7), use the same hash function (SHA256) and recalculate the digest abr of ser.

Step 6. According to formula (8), decrypt the obtained signature sig to get a new digest, which is recorded as abr$'$.

Step 7. According to formula (10), compare the values of abr and abr$'$ to judge the service integrity.

6.1.2. Experiment Analysis

(1) Confirmation of the sender's true identity: in Step 2, the sender encrypts the digest with its unique private key SK_A and gets the sender's signature sig_i, which cannot be forged. The service is delivered together with the signature. Due to the uniqueness of the signature, the sender must be responsible for the content of the service. Thus, the nonrepudiation of the sender can be guaranteed

(2) Maintenance of confidentiality: in Step 3, the cyphertext is encrypted by the receiver's public key PK_B, so only the receiver's private key can decrypt the cyphertext. Even if others intercept the cyphertext, they cannot know the content of the cyphertext. This ensures the confidentiality of the service

(3) Confirmation of the receiver's true identity: in Step 4, only the receiver's private key SK_B can decrypt the cyphertext of the receiver's public key PK_B. Once the cyphertext is decrypted, it must be the cyphertext received by the receiver to perform the decryption operation. Therefore, the decryption process itself can confirm the true identity of the receiver

(4) Integrity detection of service: the digest calculated by the service and the digest information saved in the signature is essentially calculated from the service. If the content of the service is tampered with during transmission, any small change will result in a completely different digest. Then, the new digest calculated by the receiver through the hash function must be inconsistent with the digest decrypted in the signature. Therefore, by comparing the calculated digest with the digest in the signature, whether the service has been tampered with during transmission can be confirmed, so as to the trustworthiness of the service

6.2. Experiment and Analysis of Trusted Model of Cloud Service System Runtime

6.2.1. Experiment Design. In order to verify the trustworthiness of cloud service systems based on block chain technology, this experiment is designed. This experiment focuses on the process of constructing the state chain, but there are no specific requirements for the service itself. Therefore, in order to more specifically verify the runtime trustworthiness model, our paper designed four simple services and randomly generated the message communication relationship between services, that is,

$$S = \{\mathrm{ser}_1, \mathrm{ser}_2, \mathrm{ser}_3, \mathrm{ser}_4\}, \tag{14}$$

```
st = Hash(sys_info) = "52a6cbc3f20ce4ebb293e7fcdb2722c3cbd3
04c6c4e99d41b72b87f4d2a922f4"
```

Code 4

Table 1: State chain.

State chain	Services and relationships contained	Contents of header
st	—	*52a6cbc3f20ce4ebb293e7fcdb2722c3cbd 304c6c4e99d41b72b87f4d2a922f4*
st $\longrightarrow$ Header$_1$	ser$_1$, r_1	*816c02329c388c3012961c809a69911310636 7d324a0551cb56aa4013382fee4 020b9d64225f909183d641322dbf901cef11 fc015156987e092c9f1f3445d214*
st $\longrightarrow$ Header$_1$ $\longrightarrow$ Header$_2$	ser $_1$, ser$_2$, r_2	*ef3daff6529f1874fbdd0a8219f5c246 2ef658baaba3f645d6ff4ca82b67c2cb e625ac37260f7653cf7bb2c361b81b2ce1f 77f03d1095e75ad3bbcb346003e80*
st $\longrightarrow$ Header$_1$ $\longrightarrow$ Header$_2$ $\longrightarrow$ Header$_3$	ser$_1$, ser$_2$, ser$_3$, r_3	*c52c489ca818292a6b2738efd17b83d208 7f646d48aeb1acd28890a66058b541 6df28493e4d015884fac6ca944be6f42 8981c4d4ec8cd6598a339ff3ca154734*
st $\longrightarrow$ Header$_1$ $\longrightarrow$ Header$_2$ $\longrightarrow$ Header$_3$ $\longrightarrow$ Header$_4$	ser$_1$, ser$_2$, ser$_3$, ser$_4$, r_4	*72ed72575903894c75228f7ff294ec72 2e2d57980295fe1340b7d306447240e2 853f02d65c1f026c2fc40f627fb6ebfe 4e231852e9bc583ec372f2f58071bafd*

$$Q = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \quad (15)$$

According to formula (2) and relationship matrix Q, $R = 0101001111000100$.

According to Definition 2, each service can be described as follows:

Hash function SHA256 is used during the chain building process. Suppose that the contents of sys_info is "This is the first nodes!", then, according to formula (11),

Then, the head of the state chain is built. Next, add the four services to the system in turn, and each addition or modification recreates the state block and adds it to the state chain. Then, the chain is established, which is shown in Table 1.

6.2.2. Experiment Analysis. Based on block chain technology, this paper constructs the state chain of cloud service systems. Take the construction process and modification process of cloud service systems as state information. Whether adding, deleting, or modifying system services and structures, it will actively trigger the chain building mechanism, establish new state blocks, and add them to the existing state chain. During the system runtime, any unauthorized modification in the system, whether caused by intrusion or system change caused by other reasons, will not trigger the chain building mechanism. Assuming that the status after each authorized addition, deletion, and modification is normal, the real-time status block information of the cloud service system will not match the status block modification in the chain after unauthorized modification, so the trusted status of the system can be determined. At the same time, by tracing back to the previous state in the state chain, the scope of unauthorized modification can be determined to a certain extent, so as to facilitate further error correction and improve the trustworthiness of the system.

```
"72ed72575903894c75228f7ff294e
c722e2d57980295fe1340b7d306447240e2
2efe352f56f83b3f8d7ac75af5ba555111908df6
6a0cfee65c29347f604e37aa"
```

Code 5

For example, in the current state, if the number of service 3 was tampered with to "003", by recalculating the status of the current system, the Header contents will change to

The PreHash will not change but the RootHash changed completely. By constantly deleting the services in the currently modified system, when the third service is deleted, the current system state is consistent with that in the normal chain. In this way, it can be judged that the problem lies in the third and fourth services.

7. Conclusion and Future Works

The trustworthiness of cloud service systems not only depends on each cloud service constituting the systems but also depends on the trustworthiness of the service acquisition process and running time. The trustworthiness of the acquisition process determines whether the real cloud service is obtained and whether it can be held accountable in case of service problems. The trustworthiness of running time determines whether the system maintains its normal running state during operation. Most of the tampered systems can also run, but they will get completely wrong results, which are difficult to find, resulting in huge losses. Aiming at the above two problems, this paper proposes a cloud service-trusted delivery process model based on asymmetric encryption and hash function and a cloud service system runtime trustworthiness model based on block chain technology. By using digital signature, asymmetric encryption, hash function, block chain, and other technologies, the trustworthiness of the service delivery process and trustworthiness detection at runtime are solved. Through the targeted experimental design, on the one hand, the feasibility and effectiveness of the model are verified. On the other hand, the principle and mechanism of ensuring trustworthiness are analyzed, which provides theoretical support for protecting the system rented by users on the cloud platform.

One of the deficiencies and shortcomings is that this paper focuses on the establishment of the trustworthiness models. The formal definitions of services on cloud platforms are slightly simple in content, which are difficult to cover various types of cloud services, and need to be optimized and improved in follow-up research. In addition, the existing trusted runtime model will be further studied in the future to strengthen the fault location ability of the model, so as to improve the weakness in the existing model that can only determine the approximate range of tampering.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Yunnan Fundamental Research Projects (grant nos. 202201AT070142 and 202101AT070211), the Talent Introduction Projects of Yunnan University of Finance and Economics (grant no. 2021D16), and the Scientific Research Foundation of Yunnan Education Department (grant no. 2022J0475).

References

[1] D. Guo and H. Wang, "Development review and prospect of typical forms of network computing," *Communications of the CCF*, vol. 18, no. 2, pp. 39–45, 2022.

[2] Flexera, *2020 State of the Cloud Report*, Flexera, America, 2020.

[3] X. Rong, "Research on information security in cloud computing network environment," *Network Security Technology Surgery and Application*, vol. 7, no. 7, pp. 83-84, 2021.

[4] L. Li, C. Chen, Y. Li, and J. Li, "Overview of software fault localizaion technology," *Computer Measurement and Control*, vol. 27, no. 5, pp. 1-4–121, 2019.

[5] H. Hu, D. Liu, and S. Wang, "Web ontology language OWL," *Computer Engineering and Design*, vol. 30, no. 12, pp. 1-2–147, 2004.

[6] H. Chen, J. Wang, and W. Dong, "Highly trusted software engineering technology," *Journal of Electronics*, vol. 31, no. 12, pp. 1933–1938, 2003.

[7] X. He, J. Tian, and F. Liu, "Overview of trusted cloud platform technology," *Journal of Communications*, vol. 40, no. 2, pp. 154–163, 2019.

[8] C. Shen, "Scientific concept of network security and trusted computing," *Information Technology and Network Security*, vol. 37, no. 1, p. 105, 2018.

[9] X. Yang, P. Luo, and G. Jabeen, "The concept model of software trustworthiness based on trust-theory of sociology," *Acta Electronica Sinica*, vol. 47, no. 11, pp. 2344–2353, 2019.

[10] B. Zhao, Z. Dai, S. Xiang, and W. Tao, "A method of establishing cloud platform trustworthiness analysis model," *Journal Of Software*, vol. 6, p. 17, 2016.

[11] T. Zhang, K. Zhao, M. Yang, T. Gao, and W. Xie, "Research on privacy security risk assessment method of mobile commerce based on information entropy and Markov," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8888296, 11 pages, 2020.

[12] S. Ding, F. Lu, S. Yang, and C. Xia, "A requirement-driven software trustworthiness evaluation and evolution model," *Journal of Computer Research and Development*, vol. 48, no. 4, pp. 647–655, 2011.

[13] S. Yang, S. Ding, and C. Fu, "Software trustworthiness evaluation model considering information source correlation," *Chinese Management Science*, vol. 17, no. 6, pp. 163–169, 2009.

[14] L. Gu, Y. Guo, H. Wang, Y. Zou, and B. Xie, "Runtime software trustworthiness evidence collection mechanism based on TPM," *Journal of Software*, vol. 21, no. 2, pp. 373–387, 2010.

[15] Y. Zhang, Y. Yuan, X. Liu, and X. Sun, "Evaluation method of software service trustworthiness of E-commerce website," *Computer Application Research*, vol. 37, no. 1, pp. 244-246–244-263, 2020.

[16] X. Hu, R. Jiang, M. Shi, and J. Shang, "A privacy protection model for health care big data based on trust evaluation access control in cloud service environment," *Journal of Intelligent and Fuzzy Systems*, vol. 38, no. 3, pp. 3167–3178, 2020.

[17] D. Wang and Q. Wang, "Trustworthiness evidence supporting evaluation of software process trustworthiness," *Journal of Software*, vol. 29, no. 11, pp. 178–200, 2018.

[18] R. M. C. Ratnayake and K. Antosz, "Development of a risk matrix and extending the risk-based maintenance analysis with fuzzy logic," *Procedia Engineering*, vol. 182, pp. 602–610, 2017.

[19] W. Shang and X. Xing, "ICS software trust measurement method based on dynamic length trust chain," *Scientific Programming*, vol. 2021, Article ID 6691696, 11 pages, 2021.

[20] P. Chen, X. Wang, and D. Dang, "Construction of model based on petri net and reliability analysis based on Bayes net of web

service transaction," *Journal on Communications*, vol. 39, no. S1, pp. 99–104, 2018.

[21] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized QoS prediction forweb services via collaborative filtering," in *Presented at the IEEE International Conference on Web Services (ICWS 2007)*, Salt Lake City, UT, USA, 2007.

[22] L. Yao, Q. Z. Sheng, A. Segev, and J. Yu, "Recommending web services via combining collaborative filtering with content-based features," in *Presented at the 2013 IEEE 20th International Conference on Web Services*, Santa Clara, CA, USA, 2013.

[23] Y. Zhang, C. Yin, Z. Lu, and D. Yan, "Recurrent tensor factorization for time-aware service recommendation," *Applied Soft Computing*, vol. 85, no. 6, p. 105762, 2019.

[24] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," *Decentralized Business Review*, vol. 2009, Article ID 21260, 2009.

[25] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail, in the 12th Annual International Cryptology Conference, California, USA, 2001, pp. 139-147: Springer, Berlin," in *Annual International Cryptology Conference CRYPTO 1992: Advances in Cryptology — CRYPTO' 92*, Lecture Notes in Computer Science book series (LNCS,volume 740), Springer, Berlin Heidelberg, 1993.

[26] Y. Yuan and F. Wang, "Blockchain: the state of the art and future trends," *Acta Automatica Sinica*, vol. 42, no. 4, pp. 481–494, 2016.

[27] R. Huang, "Research on supervision of financial blockchain technology," *Academic Forum*, vol. 39, no. 10, pp. 53–59, 2016.

[28] Y. Zhu, W. Song, D. Wang, D. Ma, and W. C.-C. Chu, "TA-SPESC: toward asset-driven smart contract language supporting ownership transaction and rule-based generation on blockchain," *IEEE Transactions on Reliability*, vol. 70, no. 3, pp. 1255–1270, 2021.

[29] L. Zhuang and C. Zhao, "Research on the evolution of digital currency under blockchain technological innovation: theory and framework," *The Economist*, vol. 5, no. 5, pp. 76–83, 2017.

[30] K. Li, Y. Liu, H. Wan, and Y. Huang, "A discrete-event simulation model for the bitcoin blockchain network with strategic miners and mining pool managers," *Computers & Operations Research*, vol. 134, pp. 105365–105365, 2021.

[31] J. Shi and R. Li, "Overview of blockchain access control under the Internet of Things," *Journal of Software*, vol. 30, no. 6, pp. 1632–1648, 2019.

[32] Z. Abubaker, N. Javaid, A. Almogren, M. Akbar, M. Zuair, and J. Ben-Othman, "Blockchained service provisioning and malicious node detection via federated learning in scalable Internet of Sensor Things networks," *Computer Networks*, vol. 204, pp. 108691–108691, 2022.

[33] M. Suvarna, K. S. Yap, W. Yang, J. Li, Y. T. Ng, and X. Wang, "Cyber-physical production systems for data-driven, decentralized, and secure manufacturing–a perspective," *Engineering*, vol. 7, no. 9, pp. 1212–1223, 2021.

[34] Y. Chengyue, M. Prabhu, M. Goli, and A. K. Sahu, "Factors affecting the adoption of blockchain technology in the complex industrial systems: data modeling," *Complexity*, vol. 2021, 10 pages, 2021.

[35] M. Liu, Z. Chen, Y. Shi, L. Tang, and D. Cao, "Research progress of blockchain in data security," *Chinese Journal of Computers*, vol. 44, no. 1, pp. 1–27, 2021.

[36] N. Deb, M. A. Elashiri, T. Veeramakali, A. W. Rahmani, and S. Degadwala, "A metaheuristic approach for encrypting blockchain data attributes using ciphertext policy technique," *Mathematical Problems in Engineering*, vol. 2022, 10 pages, 2022.

[37] Q. Shao, Z. Zhang, Y. Zhu, and A. Zhou, "Survey of enterprise blockchains," *Journal of Software*, vol. 30, no. 9, pp. 2571–2592, 2019.

[38] W. Yang, E. Aghasian, S. Garg, D. Herbert, L. Disiuta, and B. Kang, "A survey on blockchain-based Internet service architecture: requirements, challenges, trends, and future," *IEEE Access*, vol. 7, pp. 75845–75872, 2019.

[39] X. Han, Y. Yuan, and F. Wang, "Security problems on blockchain: the state of the art and future trends," *Acta Automatica Sinica*, vol. 45, no. 1, p. 206, 2019.

[40] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, "Applications of blockchains in the Internet of Things: a comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1676–1717, 2019.