Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.043974

ARTICLE

Blockchain-Empowered Token-Based Access Control System with User Reputation Evaluation

Yuzheng Yang*, Zhe Tu, Ying Liu and Huachun Zhou

School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, 100044, China

*Corresponding Author: Yuzheng Yang. Email: 21120151@bjtu.edu.cn

Received: 17 July 2023 Accepted: 28 October 2023 Published: 26 December 2023

ABSTRACT

Currently, data security and privacy protection are becoming more and more important. Access control is a method of authorization for users through predefined policies. Token-based access control (TBAC) enhances the manageability of authorization through the token. However, traditional access control policies lack the ability to dynamically adjust based on user access behavior. Incorporating user reputation evaluation into access control can provide valuable feedback to enhance system security and flexibility. As a result, this paper proposes a blockchain-empowered TBAC system and introduces a user reputation evaluation module to provide feedback on access control. The TBAC system divides the access control process into three stages: policy upload, token request, and resource request. The user reputation evaluation module evaluates the user's token reputation and resource reputation for the token request and resource request stages of the TBAC system. The proposed system is implemented using the Hyperledger Fabric blockchain. The TBAC system is evaluated to prove that it has high processing performance. The user reputation evaluation model is proved to be more conservative and sensitive by comparative study with other methods. In addition, the security analysis shows that the TBAC system has a certain anti-attack ability and can maintain stable operation under the Distributed Denial of Service (DDoS) attack environment.

KEYWORDS

Access control; reputation evaluation; feedback; blockchain

1 Introduction

With the development of Internet of Things (IoT) and mobile communication technologies, the Internet of Everything has accelerated the collection, analysis, and sharing of information on the Internet [1]. However, heterogeneous access needs and growing information also create serious security issues such as data leakage, information tampering, excessive access, and illegal access [2]. Therefore, it is crucial to design an access control system that can effectively protect resources.

Access control is a method of protecting resources, which usually occurs after identity authentication and plays a vital role in the protection of digital information and the maintenance of resource permissions [3]. Traditional access control is unable to provide manageable and trusted mechanisms to accommodate the heterogeneous data characteristics and short-term authorization requirements of

existing network communications [4]. Moreover, the existing centralized implementation architecture cannot meet the requirements of decentralized and transparent information management [5]. Therefore, how to build a distributed and trusted access control system has become a hot research topic.

Blockchain is defined as a distributed data structure that enables resource sharing among network nodes without relying on a central repository. Smart contracts in the blockchain can also automatically process transactions strictly according to the rules. Currently, research combining blockchain with access control systems has covered a wide range of fields such as finance [6], healthcare [7], digital information [8] and IoT, contributing greatly to the improvement of invariance, security, and accountability of access control.

In communication environments with large amounts of access, the user's reputation is also a key concern [9]. Traditional access control policy is subjective and lacks the ability to dynamically adjust based on user access behavior. Incorporating user reputation evaluation into access control can provide valuable feedback to enhance system security and flexibility [10]. However, the existing user reputation evaluation model is only based on the access control authorization process and lacks an evaluation mechanism that can evaluate the authorization and verification process.

2 Related Works

Access control is a method of ensuring that resources can only be accessed by authorized users in an authorized manner. The traditional access control methods are: discretionary access control (DAC) [11], mandatory access control (MAC) [12], role-based access control (RBAC) [13], attribute-based access control (ABAC) [14], etc. However, with the growth of resources and users, traditional access control methods cannot provide manageable mechanisms to support distributed systems with many interacting services. To solve this problem, Gusmeroli et al. [15] proposed the capability-based access control (CapBAC), which maps permission to passable capability tokens. The model can control the user's access to resources by constraining the token. Gan et al. [16] proposed the token-based access control (TBAC) model for a more standardized description of the use of tokens in access control systems. However, existing access control methods still face security problems such as single point of failure due to centralized architecture. Although a distributed solution based on lightweight IoT devices is proposed in CapBAC, it still fails to cope with malicious traffic attacks.

The distributed, decentralized, and tamper-proof characteristics of blockchain make it a natural security advantage in the field of access control [17]. Many scholars at home and abroad began to try to combine blockchain with traditional access control [18]. Sun et al. [19] proposed a blockchain-based cross-domain RBAC model using blockchain to store the mapping rules of subjects and roles and access control policies. Rouhani et al. [20] proposed a blockchain-based distributed ABAC model, which consists of an off-chain system and blockchain. The off-chain system relies on the blockchain to store its access control attributes and query access rights. Liu et al. [21] proposed a smart contract-based access control model. This model uses three smart contracts to manage the subject, policy, and authorization information respectively. Xu et al. [22] proposed a blockchain-based decentralized federated access control system BlendCAC, which uses an identity-based entitlement token management policy to register, propagate, and revoke access authorizations using smart contracts. Chen et al. [23] proposed a capability- & blockchain-based fine-grained and flexible access control model (CB2FAC), which enables granting, revoking, authenticating, and fine-grained control of user privileges through capability tokens.

In order to protect the resources in the system, user reputation evaluation needs to be used to evaluate the access behavior of users. Ghafoorian et al. [24] proposed a reputation method based

on the Bayesian reputation principle and combined it with RBAC to evaluate the trust of resource owners for roles and roles for users. Yang et al. [25] based on ABAC, used blockchain to store user behavior data. The sigmoid function and time weighting method are used to evaluate user reputation. Gwak et al. [26] proposed TARAS for the RBAC model to count the positive and negative behaviors of users and to quickly establish trust relationships between users and resources. Zhao et al. [27] proposed an attribute-based user reputation evaluation model, in which each attribute has a trust value, and the weighted average of all attributes is the total user reputation. Putra et al. [28] designed a trust-aware access control model TAC-IoT. The model evaluates user reputation based on the Gompertz function and uses an aging factor to assign different weights to access times. The paper [29] extended [28] to include the evaluations of other resources with which the user has a transaction history as a reference for reputation evaluation. Dubey et al. [30] proposed a credit component module that is integrated with ABAC and is used in cloud environments. It can be seen that most reputation models are single reputation evaluations for RBAC and ABAC models, and lack a reputation evaluation mechanism capable of evaluating the access control behavior of the user's authorization and verification process.

2.1 Motivation

Access control systems are essential for securing data and preventing unauthorized access. However, current systems have issues such as coarse-grained, poor privilege management, and centralized architecture. To solve these issues, we need standardized rules for precise and flexible privilege management, as well as implementing blockchain technology to enhance security. This can involve distributed ledgers and smart contracts to securely enforce access control policies. By doing so, we can achieve a more secure system that protects sensitive data and resources, ensuring only authorized users can access them and enhancing overall security.

Introducing the user reputation evaluation mechanism can improve the security of the access control system. However, it is still challenging to effectively combine both of them. Firstly, a reputation evaluation model needs to be designed to quantify the user's reputation level by considering the user's historical access request behavior. Then, the reputation evaluation model should have the ability to dynamically control access privileges to provide a new basis for access control. In addition, it is also necessary to design a reasonable feedback mechanism that can limit the abnormal requests of users in the authorization and verification process promptly according to the characteristics of the access control process.

Therefore, this paper aims to solve the problems of coarse-grained, poor manageability, and poor security in the existing access control. A standardized access control interaction method is developed for users and resources, and blockchain is used for distributed implementation of the system. And design a user reputation feedback mechanism that can guide access control in the authorization and verification process to restrict abnormal user behavior. The system in this paper can prevent unauthorized access and potential security threats more effectively. It can solve several current technical challenges such as data security, fine-grained access control, privilege management and reclamation, and security auditing, and can better protect data security and establish a trustworthy interaction environment between users and resources.

2.2 Contribution

The main contributions of this paper are as follows:

(1) We propose a TBAC model. This model takes the token as the credential for accessing resources and refines the traditional single policy decision to a combination of policy decision and token

decision, which improves the manageability of access control. The model realizes fine-grained access control by taking the resource operation privilege as the smallest unit.

(2) We propose a user reputation evaluation module. The model calculates the user's token reputation and resource reputation for the TBAC process, evaluates user authorization and verification, respectively, and provides feedback on access control.

(3) We use the Hyperledger fabric blockchain for the distributed implementation of the TBAC system. We performed evaluation analysis and comparative study of the TBAC system. The results show that the TBAC system can respond to access requests correctly and efficiently. The user reputation evaluation model accurately evaluates user reputation and provides feedback, demonstrating higher conservatism and sensitivity compared to previous studies [26,28,29]. We also performed the security analysis and verified the performance of the system in the Distributed Denial of Service (DDoS) attack environment.

2.3 Organization

The remainder of this paper is organized as follows. In Section 3, we present background information on access control and blockchain. In Section 4, the system framework and main modules are described. In Section 5, we describe the process of TBAC and user reputation evaluation and introduce the update and feedback mechanism of user reputation. In Section 6, we evaluate the TBAC system and the reputation model. Performed security analysis and verified the performance of the system in the DDoS attack environment. In Section 7, we conclude this paper and introduce future work.

3 Preliminaries

3.1 Access Control

Access control is a technique and methodology used to manage and control access to resources in a computer system, network, or physical location [31]. It determines who can access specific resources, under what conditions, and in what manner. The primary objective of an access control system is to protect the security and confidentiality of resources, preventing unauthorized access, misuse, and data leaks. By implementing features such as authentication, authorization management, and auditing, access control systems verify the identity of users or entities and determine their access privileges. This ensures that only authorized individuals or entities are granted appropriate privileges to access restricted resources, thereby maintaining the overall security and integrity of the system. In this paper, the TBAC system uses a token as a permission credential to the smallest unit of resource operation authority, which can realize manageable fine-grained access control.

3.2 Blockchain

Blockchain is regarded as an immutable distributed database [32]. It integrates networks, cryptography, game theory, and smart contracts to facilitate the entire process from bookkeeping to distribution, verification, and preservation. Blockchain is classified into three types: public chain, private chain, and consortium chain. The public chain is a completely open blockchain network, where the data of the public chain is transparent and public so that anyone can view and verify it. The private chain is a restricted blockchain network where only specific organizations or entities are authorized to join. The consortium chain is a type of blockchain network that is jointly managed and verified by multiple organizations or entities. The participants jointly decide and manage the network. In order to

ensure higher security and flexibility in the network environment, this paper adopts a representative Hyperledger Fabric in the consortium chain for the distributed implementation of the system.

4 System Overview

In this section, we first introduce the system framework, and then introduce the TBAC module, the user reputation evaluation module, and the blockchain module, respectively.

4.1 TBAC System Framework

As shown in Fig. 1, the TBAC system consists of three parts: the TBAC module, the user reputation evaluation module, and the blockchain module.

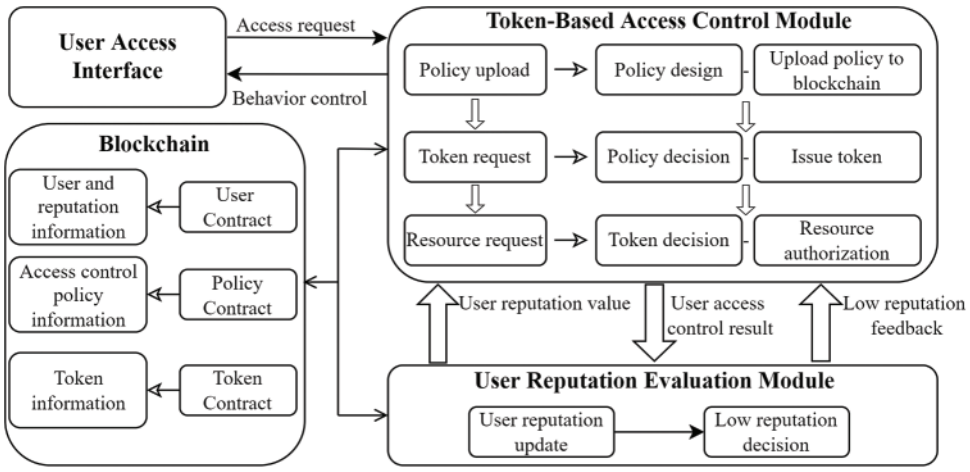


Figure 1: TBAC system framework

The TBAC module is the main module of the system. The module receives and processes access control requests from users and controls their access behavior. The TBAC module divides the access control process into three stages: policy upload, token request, and resource request.

The user reputation evaluation module provides reputation evaluation for the access control module. It evaluates and updates the reputation based on the user's access control results. This module provides user reputation for the access control decision process and provides feedback to abnormal users through low reputation decisions based on the user reputation update.

The system uses blockchain for distributed implementation. The blockchain contains three types of smart contracts: User Contract, Policy Contract and Token Contract, which are used to implement the decision process, and user reputation evaluation process in the access control.

4.2 Token-Based Access Control Module

The TBAC module divides the access control process into three stages: policy upload, token request, and resource request, as shown in Fig. 2.

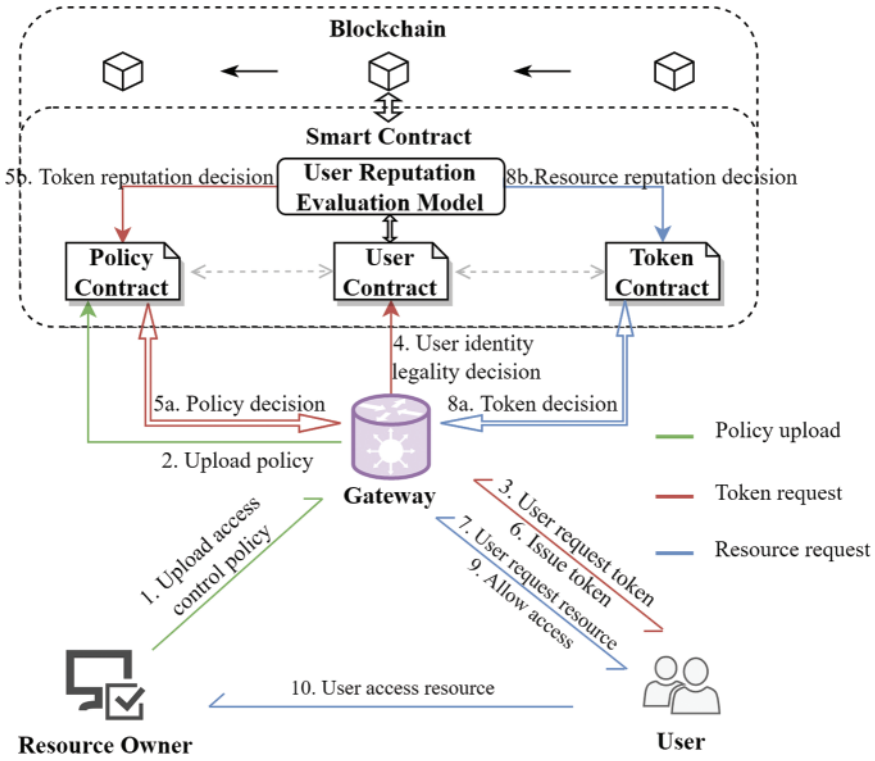


Figure 2: The process of TBAC model

In the policy upload stage, the resource owner uploads the predefined access control policy to the blockchain through the gateway for the subsequent decision process of user access requests. In the token request stage, the user submits token request (3). The gateway invokes the blockchain smart contract to query the legality of the user's identity (4) and make policy decision (5a). In the process of policy decision, the system invokes the user reputation evaluation module for token reputation decision (5b). If the policy decision is satisfied, a token is issued for the user (6), which can be used as a credential for the user's resource access. In the resource request stage, the user continues to submit a resource request (7), and the gateway invokes the smart contract for token decision (8a). The system also invokes the user reputation evaluation module for resource reputation decision (8b). After the token decision, the user can be granted the corresponding resource access rights (9) based on the valid information of the token. Then the user can start to access the resource (10).

4.3 User Reputation Evaluation Module

In order to improve the security and flexibility of the system, this paper uses the user reputation evaluation module to evaluate the user's token reputation and resource reputation and provide reputation information and feedback for access control, respectively. The process of the module is shown in Fig. 3.

In the policy decision of the token request (1), the reputation model is invoked to query the token reputation and make the token reputation decision (2). At the end of the token request, the reputation model updates the token reputation based on the token request result (3). Then make low reputation decision, if the reputation is not satisfied, the low reputation feedback will be triggered

and the system will limit the user's identity legality (4). In the token decision of the resource request (5), the User Contract is invoked for the resource reputation decision (6). At the end of the resource request, the user's resource reputation is updated based on the resource request result (7). Similarly, the system makes the low reputation decision and triggers low reputation feedback if the reputation is not satisfied, and the system will limit the validity of the user's token (8).

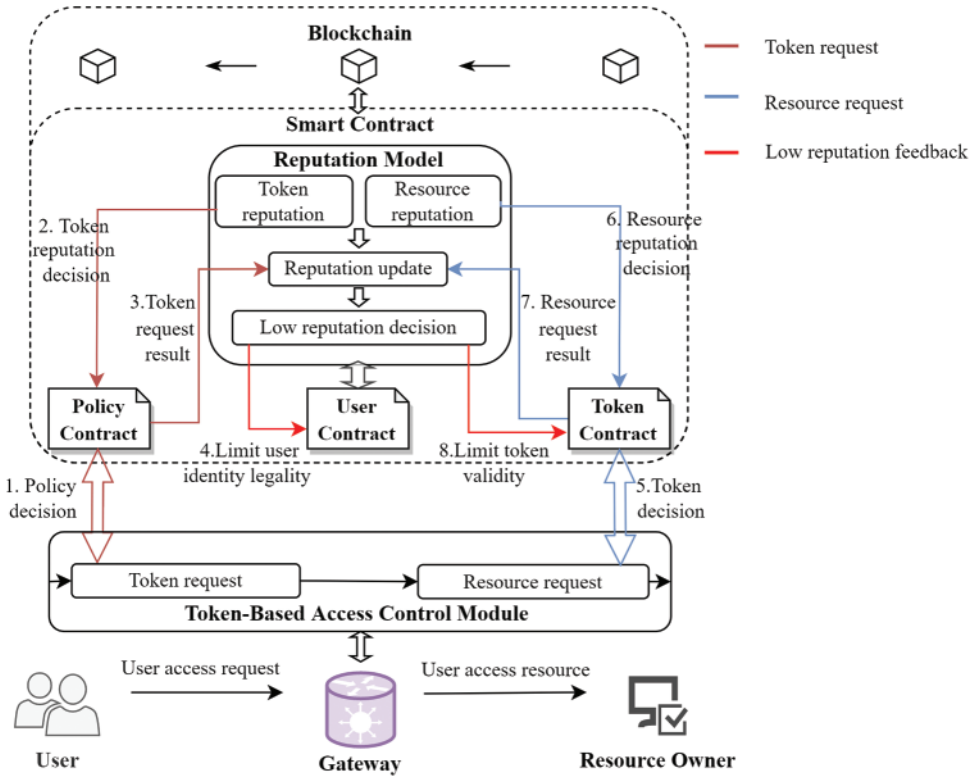


Figure 3: The process of user reputation evaluation module

4.4 Blockchain Module

In this paper, three smart contracts are deployed in the blockchain: User Contract, Policy Contract, and Token Contract, which are used to store and process user and reputation information, access control policy information, and token information respectively. Meanwhile, we implement access control and reputation evaluation by defining smart contract functions. The smart contracts and their functions are described in Table 1.

In Table 1, the *UID*, *OID*, *RID* and *TID* are the identifiers of the user, resource owner, resource and token, respectively. *Policy* is the access control policy, $Policy := \{RID, OP, < UID >, < Role >, < IP >, < Loca >, Period\}$, in which, *OP* indicates an operation allowed for the resource. The remaining fields indicate the conditions to be satisfied to access the resource: $< UID >$ is the set of limited user identifiers, $< Role >$, $< IP >$, and $< Local >$ are the set of roles, IP addresses, and geographic locations of permitted users, respectively; *token* represents the access control token, $token := \{TID, UID, OID, RID, OP, Period, ET, N, VI\}$, in which *ET* is the token expiration time, N is the usage times of the token; and VI is the token validity identifier; *TRR* and *RRR* are the token request result and resource request result, respectively.

Table 1: Smart contract and function description

Smart contract	Main function	Function description
User contract	*QULegality* (*UID*)	Return the user's current identity legality.
	QueryUTR (*UID*, *OID*)/*QueryURR* (*UID*, *OID*)	Returns the user's reputation for reputation decision.
	UpUTR (*TRR*)/*UpURR* (*RRR*)	Update user reputation, make low reputation decision and feedback.
Policy contract	*UpPolicy* (*Policy*)	Upload access control policy on the blockchain.
	PolicyDec (*UID*, *OID*, *RID*, *OP*, *Role*, *IP*, *Loca*)	Make policy decision in the token request.
Token contract	*UpToken* (*token*)	Upload access control token on the blockchain.
	TokenDec (*UID*, *OID*, *RID*, *OP*, *TID*)	Make token decision in the resource request.
	TokenINV (*UID*, *OID*)	Invalidate all tokens of the user for the resource owner.

5 Token-Based Access Control System

In this section, we first introduce the specific process of the system and then describe the principles of the reputation model.

5.1 Token-Based Access Control

The specific process of each stage of TBAC is shown in Fig. 4.

(1) Policy Upload

Step 1: The resource owner sends the upload policy message UPM to the gateway, UPM:= {Policy}.

Step 2: The gateway invokes the *UpPolicy*() function of the Policy Contract for policy upload. The blockchain executes the chain code function to upload the access control policy.

(2) Token Request

Step 3: The user requests the token. The user submits the identity legality verification request message IRM. $IRM := \{UID, PK_A\}$, in which PK_A is the user's public key.

Step 4: User identity legality decision. 4.1. The gateway invokes the *QULegality*() function in the User Contract to make the user identity legality decision to get the identity legality result IRR. 4.2. The gateway returns to the user the identity legitimacy decision result message $IRRM$, $IRRM := \{IRR, PK_B\}PK_A$, which includes IRR and the gateway public key PK_B, encrypted with PK_A.

Step 5: Policy decision. 5.1. The user sends the token request message TRM to the gateway. $TRM := \{SignSK_A[UID, OID, RID, OP, Role, IP, Loca]\}PK_B$, which contains the user information

corresponding to the policy, signed by SK_A and encrypted by PK_B. 5.2–5.3. Invoke *PolicyDec*() function in the Policy Contract for policy decision. 5.4. Make token reputation decision in the policy decision. 5.5. Generate the access control token *token* for the user after the policy decision. Invoke the *UpToken*() function in the Token Contract to upload the token and save it. 5.6–5.7. Get the *TRR* and invoke the *UpUTR*() function in the User Contract to update the user token reputation *UTR* according to *TRR*.

Step 6: The system issues the token to the user. 6.1. The Policy Contract transmits the token generated for the user to the gateway. 6.2. The gateway returns the *TRR* message *TRRM* to the user, $TRRM := \{TRR, TID, UID, OID, RID, OP, Period\}PK_A$, which includes *TRR* and part of the token information. 6.3. The user extracts the token information from *TRRM* and then can make resource request via the token.

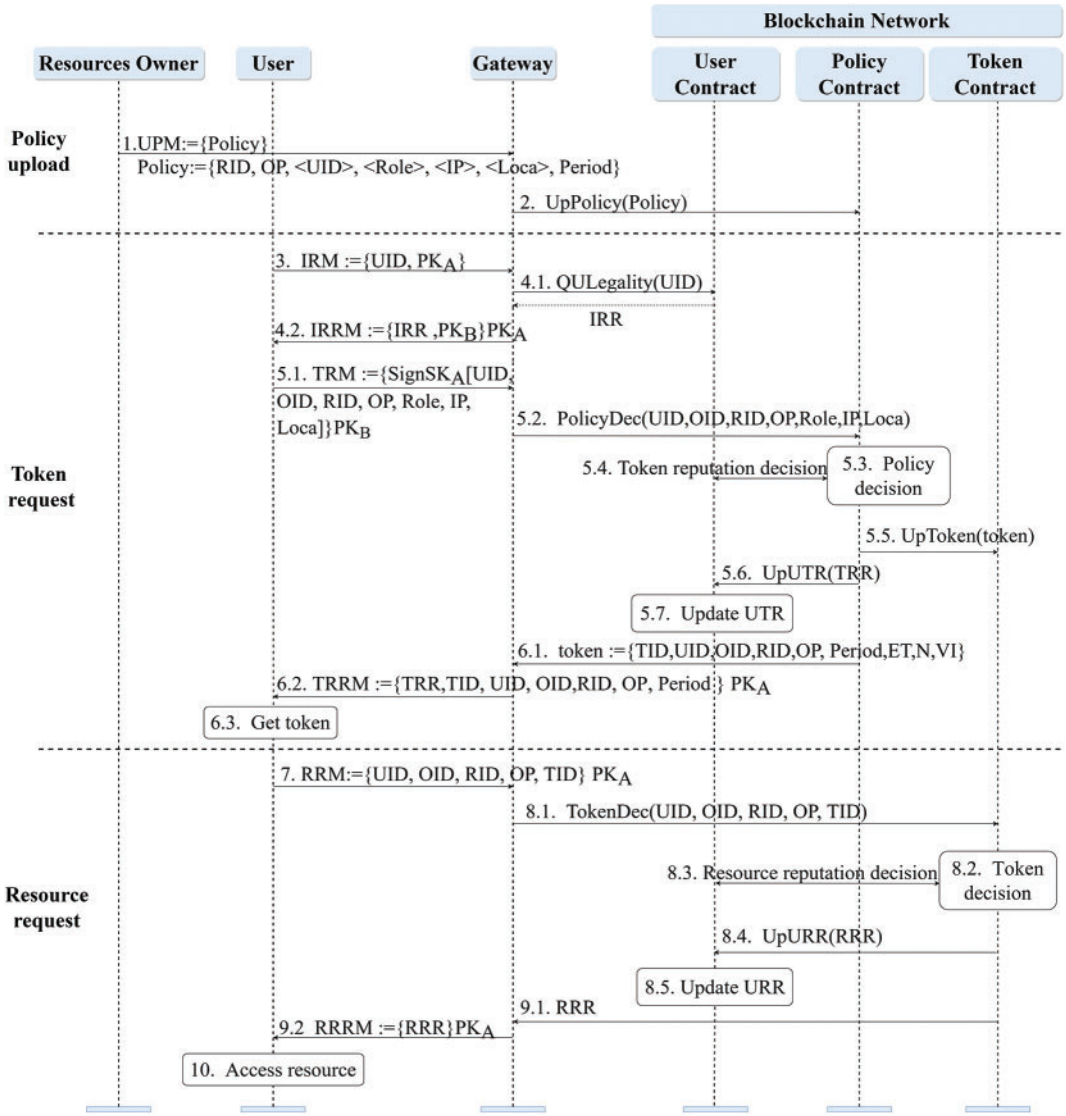


Figure 4: The specific process of each stage of TBAC

(3) Resource Request

Step 7: The user requests resource. The user sends a resource request message RRM to the gateway, $RRM := \{UID, OID, RID, OP, TID\}PK_A$.

Step 8: Token decision. 8.1–8.2. Invoke the *TokenDec*() function in the Token Contract to make token decision. 8.3. Make resource reputation decision in the token decision process. 8.4–8.5. At the end of the decision, the *UpURR*() function of the User Contract is invoked to update the URR based on the RRR.

Step 9: Return the RRR to the user. 9.1. The Token Contract returns the RRR to the gateway. 9.2. The gateway generates the resource request result message $RRRM$, $RRRM := \{RRR\}PK_A$, and forwards it to the user.

Step 10: The user accesses the resource after obtaining the permission.

5.2 User Reputation Evaluation

In this subsection, we will describe the specific process of the user reputation evaluation and the principle of user reputation update and feedback mechanisms.

5.2.1 Process of User Reputation Evaluation

The specific process of each stage of user reputation evaluation is shown in Fig. 5.

(1) Token request

Step 1: Policy decision in the token request. 1.1–1.2. During the token request of TBAC, the gateway invokes the *PlicyDec*() function of the Policy Contract to make policy decision.

Step 2: Token reputation decision. 2.1. As a part of the policy decision, the token reputation decision needs to invoke the *QueryUTR*() function of the User Contract to query the user's direct token reputation UTR_DR. 2.2. Make the token reputation decision based on UTR_DR.

Step 3: Token reputation update. 3.1–3.2. Invoke the *UpUTR*() function of the User Contract after the policy decision to update the UTR according to TRR.

Step 4: Low reputation decision and feedback. 4.1. Make low reputation decision after UTR update. 4.2. If the reputation is not satisfied, low reputation feedback is triggered to limit the legality of the user's identity. 4.3–4.8. The effectiveness of low reputation feedback is reflected in the user's next token request, where the system rejects the user's request due to illegal identity in the user identity legality decision.

(2) Resource request

Step 5: Token decision in the resource request. 5.1–5.2. During the resource request of TBAC, the gateway invokes the *TokenDec*() function of the Token Contract to make token decision.

Step 6: Resource reputation decision. 6.1. As a part of the token decision, the *QueryURR*() function of the User Contract is invoked to query the user's resource reputation URR. 6.2. Make resource reputation decision according to URR.

Step 7: Resource reputation update. 7.1–7.2. Invoke the *UpURR*() function of User Contract at the end of the token decision to update URR according to RRR.

Step 8: Low reputation decision and feedback. 8.1. Make low reputation decision after URR is updated. 8.2–8.3. If the reputation is not satisfied, the low reputation feedback is triggered and the *TokenINV*() function in the Token Contract is invoked to invalidate the user's token for the resource

owner. 8.4–8.8. The effectiveness of the low reputation feedback is reflected in the user's next resource request. The system rejects the user's request for the invalid token in the token decision.

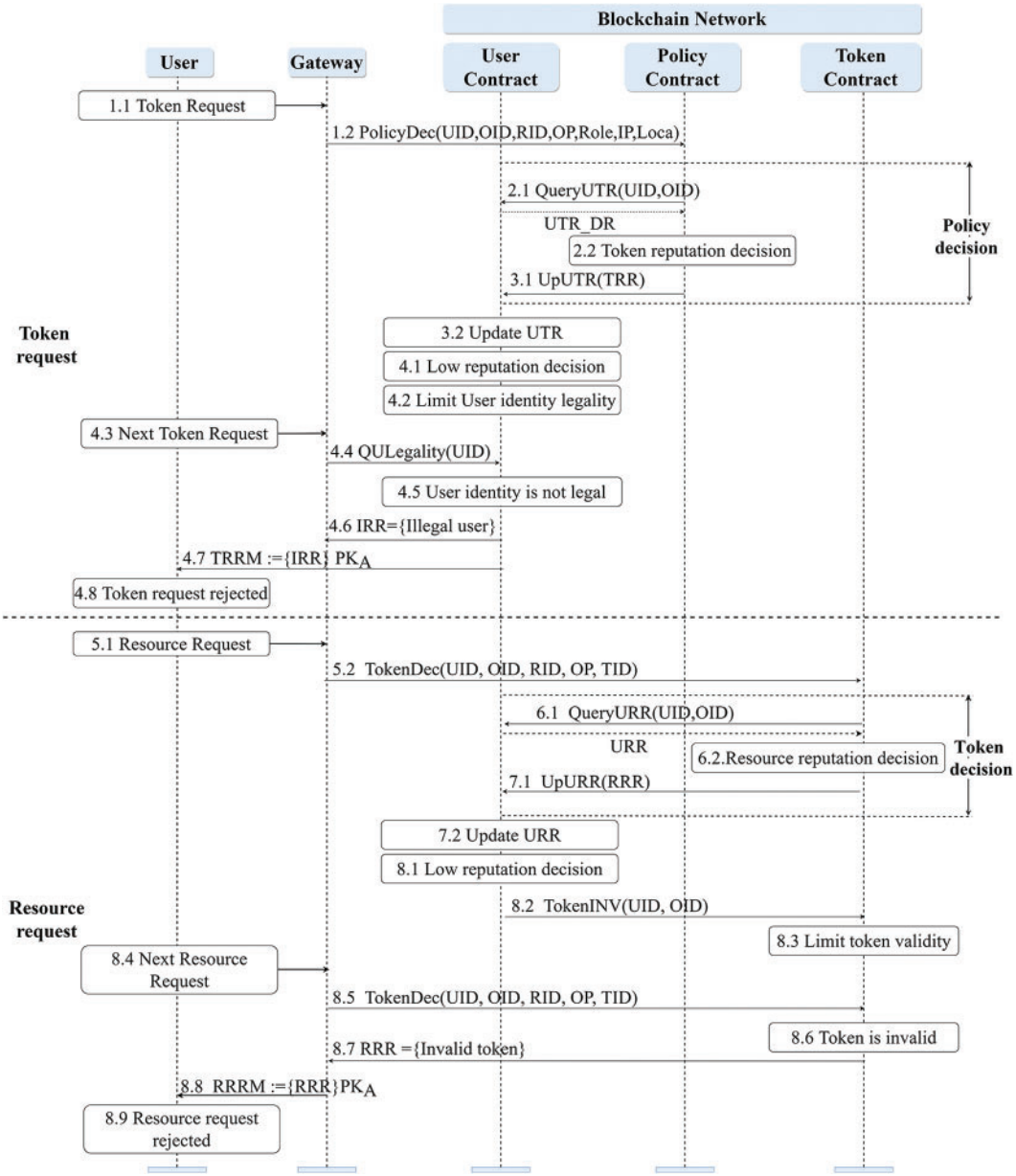


Figure 5: The specific process of each stage of user reputation evaluation

5.2.2 Reputation Model

The user reputation evaluation module evaluates the token reputation and resource reputation based on the user's access control result in the token request and resource request stages, respectively.

Reputation Update Mechanism

(1) Token Reputation

The token reputation *UTR* is the user reputation calculated based on the user's token request result, which consists of direct token reputation and recommended token reputation.

1) Direct Token Reputation

The direct token reputation *UTR_DR* is a reputation calculated directly based on the user's access request result in the token request. Based on the result, the evaluation of the user at that stage can be divided into two intervals: R $Reject \in (0, 0.5]$ and $Trust \in (0.5, 1]$. The token request result and its corresponding evaluation interval are shown in Table 2. A feedback quantified value f_T is randomly generated according to the evaluation interval, which represents the evaluation of the token request behavior of the user by the resource owner. Calculate the user direct token reputation *UTR_DR* according to Eq. (1).

$$\begin{cases} If\ f_T > 0.5, \quad then\ \ \alpha_T{}^{(n)} = \alpha_T{}^{(n-1)} + f_T - 0.5 \\ If\ f_T < 0.5, \quad then\ \ \beta_T{}^{(n)} = \beta_T{}^{(n-1)} - f_T + 0.5 \\ otherwise, \quad do\ nothing \\ UTR\_DR = \dfrac{\alpha_T{}^{(n)}}{\alpha_T{}^{(n)} + P_T \beta_T{}^{(n)}} \end{cases} \tag{1}$$

Table 2: Access control results and the corresponding evaluation intervals

TBAC stage	Access control result	Evaluation interval
Token request	Illegal user	/
	Failure to satisfy token reputation determination	$Reject \in (0, 0.5]$
	Mismatch with policy	
	Token request successful	$Trust \in (0.5, 1]$
Resource request	Failure to satisfy resource reputation decision	/
	Invalid token	$Reject \in (0, 0.25]$
	Token not exist	
	Not the token owner	
	Mismatch with token	
	Not in the access period	$SuspectReject \in (0.25, 0.5]$
	Resource request successful	$Trust \in (0.5, 1]$

In Eq. (1), n is the current request status, α_T is positive feedback, and β_T is negative feedback. In order to increase the monitoring of the abnormal access, a constant P_T greater than 1 is introduced as a penalty factor. In the first calculation of a new *UTR*, make the initial value $\alpha_T = \beta_T = 1$. When the *UTR_DR* is lower than 0.5, P_T increases by ΔP.

2) Recommended Token Reputation

Recommended token reputation *UTR_IR* is the user reputation calculated based on the recommended value of the recommender, where the recommender refers to other resource owners with whom the user has interaction history. When calculating the recommended reputation, N_R recommenders are selected first, and then the recommended reputation is calculated based on their recommended value for the user, as shown in Eq. (2).

$$UTR\_IR = \frac{\sum_{i=1}^{N_R} (T_i + D_i) * DR_i^{rec}}{\sum_{i=1}^{N_R} (T_i + D_i)} \tag{2}$$

In Eq. (2), T and D are the recommender reliability weight factors, which are used to select N_R reliable recommenders. T is the transaction reliability weight, the greater the total number of tokens issued to the user, the greater the value of T. D is the time reliability weight, the closer the time to generate the latest token, the greater the value of D. DR_i^{rec} is the recommended evaluation value, which is the direct token reputation of the recommender for the user. When the number of recommenders in the system is less than N_R, virtual recommenders are introduced to make the number of recommenders reach N_R, and will set $DR_i^{rec} = 0.5$, and the T and D factors are taken to be relatively minimal.

3) Token Reputation

The token reputation *UTR* is calculated from the weighted sum of the *UTR_DR* and the *UTR_IR*, where w is the weight of the *UTR_DR*, as shown in Eq. (3).

$$UTR{=}w * UTR\_DR + (1 - w) * UTR\_IR \tag{3}$$

(2) Resource Reputation

The resource reputation *URR* is the user reputation calculated based on the user's resource request result, which reflects the direct trust of the resource owner to the user. Based on the result, the system's evaluation of the user at this stage can be divided into three intervals: *Reject* $\in (0, 0.25]$, *Suspect Reject* $\in (0.25, 0.5]$, and *Trust* $\in (0.5, 1]$, and the resource request result and its corresponding evaluation interval are shown in Table 2. A feedback quantified value f_R is randomly generated within the evaluation interval, which represents the accessed resource owner's evaluation of that user's resource request behavior. Then, the *URR* is calculated according to Eq. (4).

$$\begin{cases} If\ f_R > 0.5, \quad then\ \alpha_R{}^{(n)} = \alpha_R{}^{(n-1)} + f_R - 0.5 \\ If\ f_R < 0.5, \quad then\ \beta_R{}^{(n)} = \beta_R{}^{(n-1)} - f_R + 0.5 \\ otherwise, \quad do\ nothing \\ URR = \dfrac{\alpha_R{}^{(n)}}{\alpha_R{}^{(n)} + P_R \beta_R{}^{(n)}} \end{cases} \tag{4}$$

when calculating the *URR* for a new user for the first time, make the initial value $\alpha_R = \beta_R = 1$. When the user reputation is below 0.5, P_R increases by ΔP.

Reputation Feedback Mechanism

After the user's reputation is updated, the user reputation evaluation module provides low reputation feedback through low reputation decisions. In this paper, four reputation thresholds are defined for these decision processes:

Access Permission Threshold (*APT*): By limiting the *UTR_DR*, the user's level of compliance with the resource owner's policy is judged in the token reputation decision.

Resource Authorization Threshold (*RAT*): By limiting the *URR*, the user's token usage for a resource is judged in the resource reputation decision.

Identity Legitimacy Threshold (*ILT*): By limiting the *UTR*, the user's level of compliance with the policy within the system is judged in the low reputation decision.

Invalid token threshold (*ITT*): By limiting the number of times the user is invalidated for a resource owner, the user is judged for the token usage of the resource owner in the low reputation decision.

As shown in Fig. 6, In the token request stage compare *UTR_DR* with *APT*. If *UTR_DR* is lower than *APT*, deny this access request. Then compare *UTR* with *ILT* in the user reputation evaluation. If *UTR* is lower than *ILT*, limit the user identity legality. The next token request will be denied because the user identity is not legal. After the penalty time has elapsed, the user's identity is restored. In the resource request stage, if *URR* is less than *RAT*, the current token is invalidated and the user is denied access. The user has to go through the token request again to reapply for a new token. In the reputation evaluation, the number of invalid tokens is counted and compared with *ITT*. If it is higher than *ITT*, it is reasonable to assume that the user is a threat to the resource owner. Then the system invalidates all the user's tokens for the resource owner, and all subsequent accesses to the resource owner by the user need to make the token request again.

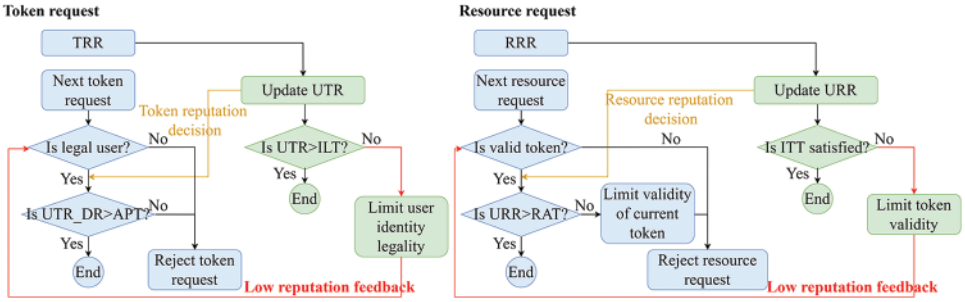


Figure 6: Reputation feedback

6 Experimental Evaluation

6.1 Experimental Setup Detail

We use the Hyperledger Fabric for a distributed implementation of the system. The network topology is shown in Fig. 7. The evaluation is performed on the VMware vSphere virtual platform, on which the virtual machines are configured as Intel® Xeon® CPU E52609 v4 @1.70 GHz ∗8, 8 GB RAM, 1 TB HDD. We use 10 virtual machines to simulate users, resource owners, and gateways respectively. The gateway is used for responding to access control requests and also as a blockchain node for transaction sequencing and ledger maintenance.

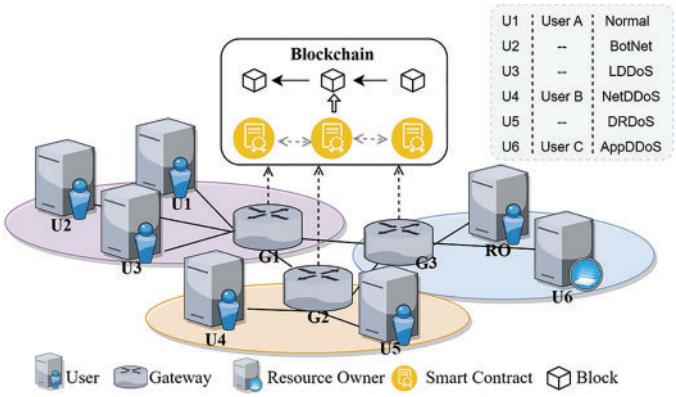


Figure 7: The network topology

In the system evaluation, we first test the delay of the system under different conditions. We define the request delay as the time spent by the user in the corresponding stage from the submission of the request to the receipt of the request result. Define the access delay as the time spent by the user in the process of submitting the token request to access the resource.

The user reputation is then evaluated, and we simulate three users, User A(U1), User B(U4), and User C(U6), respectively. Each user sends 100 token requests and resource requests. User A continuously sends normal requests, User B continuously sends abnormal requests, and User C initially sends normal requests until the 30th request is followed by an abnormal request. In addition, we compare this paper's model with existing models [26,28,29] by evaluating the reputation of User A and User B.

Finally, we integrate the TBAC model with the security feedback module [33] to test the performance of TBAC in the DDoS attack environment. The security feedback module is used to feed information about malicious users to TBAC. We then used Python scripts to simulate normal traffic on U1 and replayed five DDoS attack traffic through TcpReplay on other User VMs to simulate the DDoS attack environment [34], including BotNet (U2), LDDoS (U3), NetDDoS (U4), DRDoS (U5), and AppDDoS (U6).

6.2 Evaluation and Discussion

6.2.1 Evaluation of the TBAC System

In this subsection, we first evaluate the access control request delay in the TBAC system. Then we compare the system processing performance under different situations.

As shown in Fig. 8, the single request delay of most of the token requests and resource requests is within 0.15 and 0.05 s, respectively, which proves the high performance of the TBAC system. The higher delay of the phase is caused by the blockchain operations such as packing, consensus, and publishing the ledger.

As shown in Fig. 9, we test the total request delay required by the system to respond to 200 token requests and resource requests for different proportions of normal requests respectively. It can be found that when the proportion of normal requests decreases, the request delay also decreases. This is because the system reduces the decision delay by rejecting abnormal requests in a timely manner.

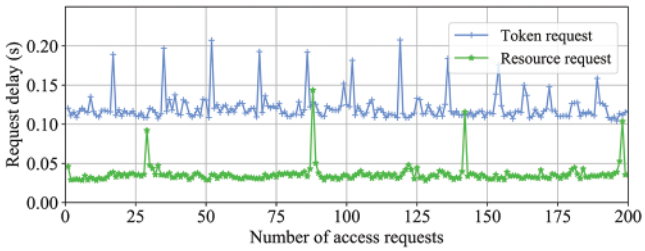


Figure 8: The request delay of token request and resource request

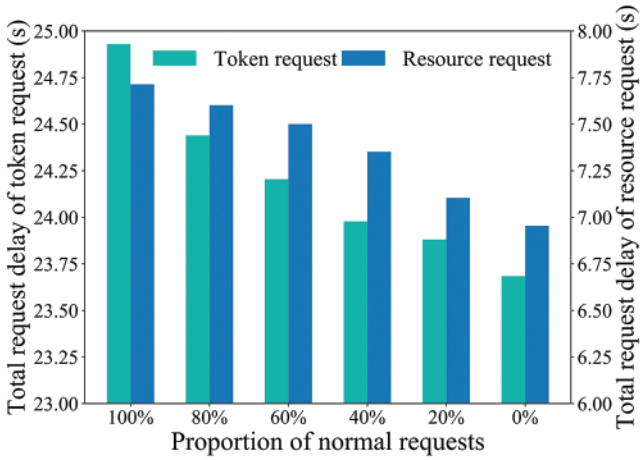


Figure 9: The total request delay for different proportion of normal requests

Then we compared the total access delay of TBAC and the Non-token system. As shown in Fig. 10, the performance of TBAC is higher, and the effect is more obvious with more accesses. This is because TBAC achieves the separation of authorization and verification through the token, reducing the overall number of request decisions in the system.

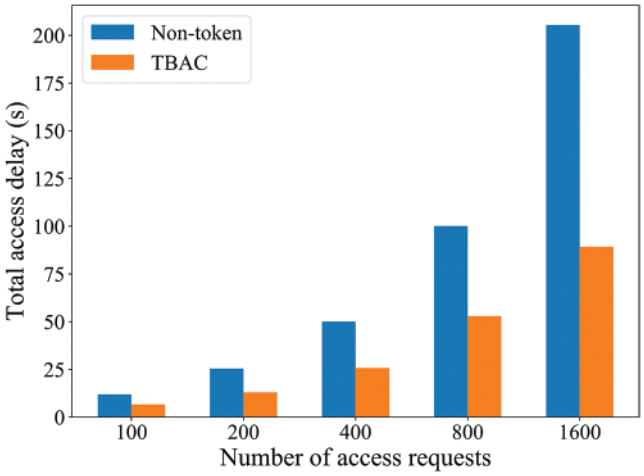


Figure 10: The total access delay of TBAC and non-token system

6.2.2 Evaluation of User Reputation

In this subsection, we first evaluate the reputation of different types of users in the access process. Then we validated the feedback from user reputation on the access control process. Finally, we compare this reputation model with other methods. In the evaluation, the parameters of the reputation model are set as $\alpha_T = \beta_T = \alpha_R = \beta_R = 1$, $P_T = P_R = 1.3$, $\Delta P = 0.3$, $N_R = 4$, $DR_i^{rec} = 0.5$, $w = 0.7$. The reputation thresholds are set as $APT = ILT = RAT = 0.3$, ITT is 2/3 of the number of tokens.

As shown in Fig. 11, we evaluate the reputation of different types of users. It can be seen that the initial value of reputation is 0.5. Normal behavior increases user reputation, as seen after the 100th request, the UTR of User A is 0.82 and the URR is 0.96. The slow increase of UTR is due to the effect of the recommended reputation. It proves that the system is conservative in rewarding reputation. Abnormal behavior leads to a decrease in reputation and low reputation feedback is triggered when it falls below the threshold (0.3). For token requests, low reputation feedback limits user identity legitimacy for a certain period of time and *UTR* stays at the default value. For resource requests, triggering low reputation feedback limits token validity and *URR* resets to the default value, and users need to make the token request again.

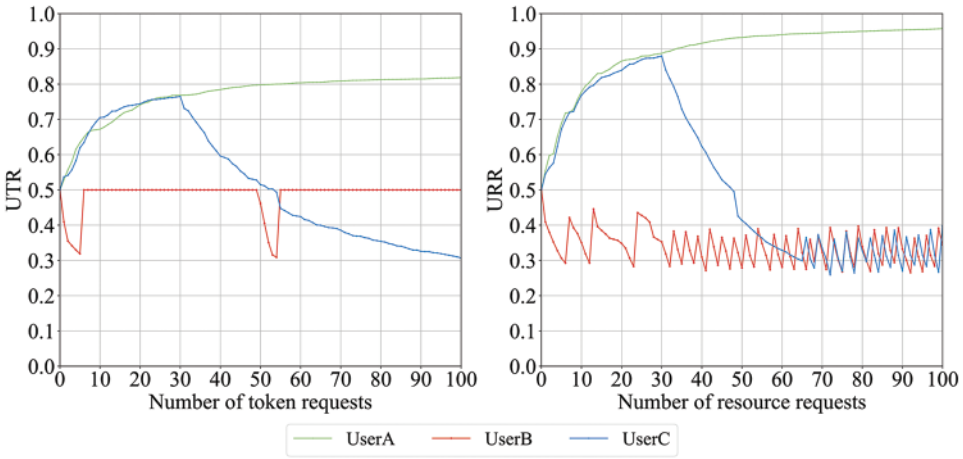


Figure 11: Reputation evaluation of different types of users

To verify the feedback of user reputation on access control, we tested the changes of access results with reputation for three users, as shown in Fig. 12. We divide the access control results into three types: successful access, failed access, and limited access, where limited access corresponds to the case where user identity legality is limited in the token request and token validity is limited in the resource request. As shown in the figure, the *UTR* and *URR* of User A are growing and all its access requests are successful. User B's all access requests failed and access was limited after low reputation feedback. User C's first 30 requests are successful and gets the failed result for the subsequent requests. In addition, during the resource request, User C triggers the low reputation feedback at the 67th request, and the access is limited. All together, the user reputation evaluation module can provide timely feedback for access control.

Then we compared the reputation model with the methods in [26,28,29]. Since other methods do not have a reputation for resource access, we compare token reputation with other models, because they both reflect the user's compliance with the access control policy. Fig. 13 shows the reputation evaluation of User A and User B in different reputation models. As shown in Fig. 13a, all reputation

curves of User A show an increasing trend. Among them, TBAC achieves a reputation value of 0.82 after 100 requests, while the other three models achieve a reputation value of more than 0.9 after the 20th request. TBAC converges significantly slower than the other models, which have higher conservativeness and stability. As shown in Fig. 13b, the reputation curves of User B are all trending downward. The reputation of TBAC decreases to below 0.3 after the 4th request, which is the larger rate of decrease than the TRS-IoT and BCTRA models, and only second to the TARAS model. Taken together, the reputation model of TBAC is highly sensitive to abnormal behavior while conservatively rewarding normal behavior, and can effectively maintain the stability and fairness of the whole system.

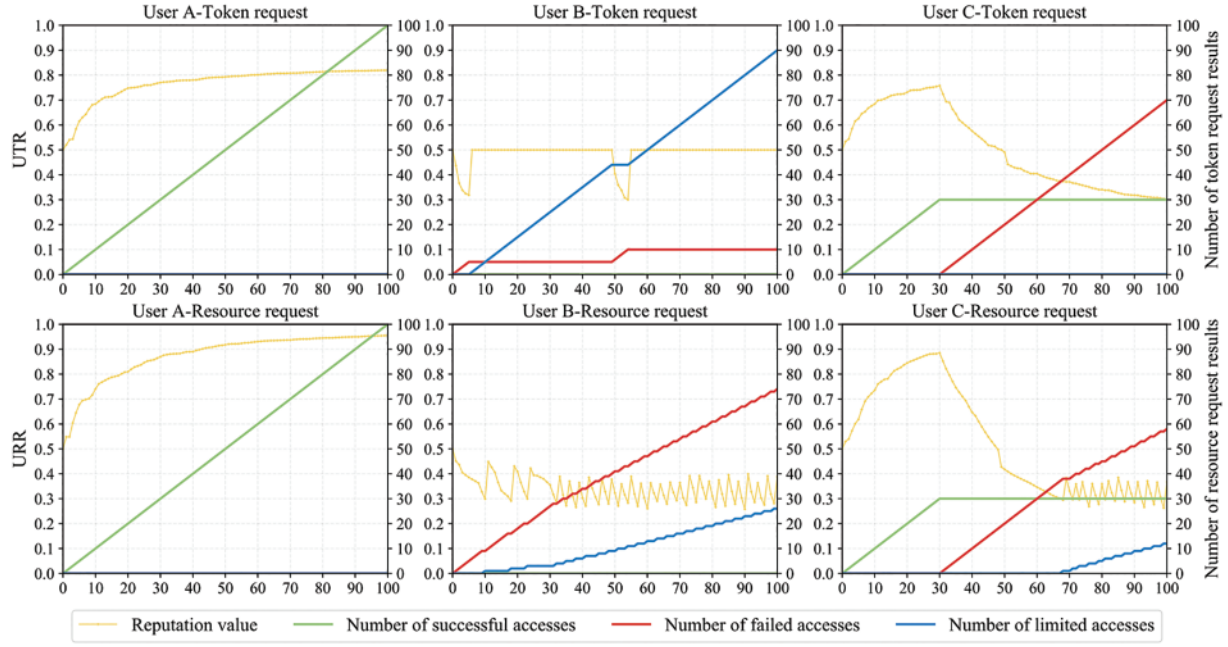
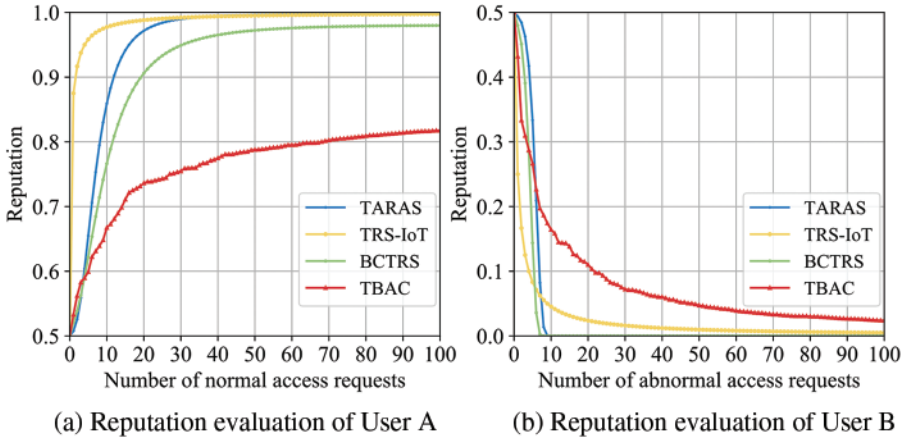


Figure 12: The changes of access results with reputation for three users



(a) Reputation evaluation of User A (b) Reputation evaluation of User B

Figure 13: Comparison of reputation methods

6.3 Security Analysis

The TBAC model does not introduce new security issues. We provide a security analysis of several common attacks that the system can resist. The TBAC model is then combined with a security feedback module to test the performance of TBAC in the DDoS attack environment.

Ultra vires attack resistant: Traditional access control allows a user to execute all the privileges it has when the request passes, which may lead to ultra vires attacks. The TBAC system in this paper authorizes the user's resource operation privileges according to the user's specific request, which effectively resists ultra vires attacks through fine-grained authorization.

Identity spoofing attack resistant: An identity spoofing attack refers to an attacker forging a legitimate user identity to gain access to resources. In this regard, the system in this paper adopts the following two measures: firstly, using blockchain to record and audit user behaviors. Secondly, using the determination policy to ensure the user's access is legitimate, and to detect and block abnormal access behaviors such as logging in from a different place, logging in at an illegal time, and holding illegal tokens in a timely manner.

Opportunistic attack resistant: Opportunistic attackers usually do not plan their attacks in advance but rather look for opportunities to exploit to carry out malicious behaviors. For example, the attacker uses the privileges obtained from previous normal requests to carry out illegal behaviors. In response, TBAC recovers privileges in a timely manner by limiting the number of times and duration of token usage to avoid over-exploitation of privileges.

Whitewashing attack resistant: Malicious users attempt to be removed from the system by lowering their reputation and then re-enter the system with a new reputation. In this regard, TBAC employs the following strategies: First, the blockchain is used to ensure that the user's history is fully recorded, and the user's bad behavior can be verified by the system at any time. In addition, the system restricts the access of such users for a certain period of time through low reputation feedback, thus preventing the whitewashing behavior.

Next, we perform an experiment to evaluate the performance of TBAC in the DDoS attack environment. We set up one user who sends normal traffic and five users who send attack traffic, each sending access control requests every 5 s with a duration of 200 s. The user sending normal traffic sends normal requests continuously. Users sending attack traffic send normal requests for the first 50 s and random normal or abnormal requests for the next 150 s. Finally, the malicious user information is fed back to TBAC via the security feedback module at 180 s.

The change in user reputation is shown in Fig. 14. As can be seen, the reputation of normal traffic users continues to increase. Attack traffic users change according to their access control behavior, where users sending BotNet and NetDDoS trigger low reputation feedback due to too many abnormal requests. after 180 s, the reputation of attack traffic users both returns to the default value. The traffic inflow at the resource owner is shown in Figs. 15, and 15a shows the traffic inflow by type. It can be seen that each type of traffic inflow varies according to the access control results and reputation changes. After the low reputation feedback is triggered (110 and 130 s), BotNet and NetDDoS traffic no longer flows to the resource. Fig. 15b shows the inflow of normal traffic and attack traffic more visually. As can be seen, after 180 s, TBAC receives the security feedback module malicious user information and is able to block malicious users, thus blocking malicious traffic.

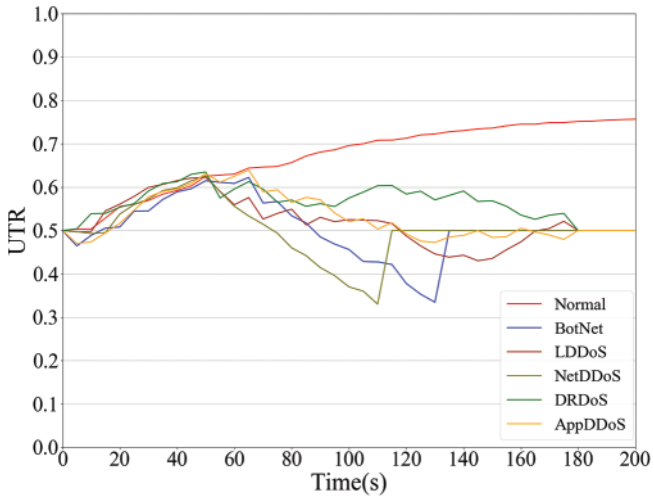
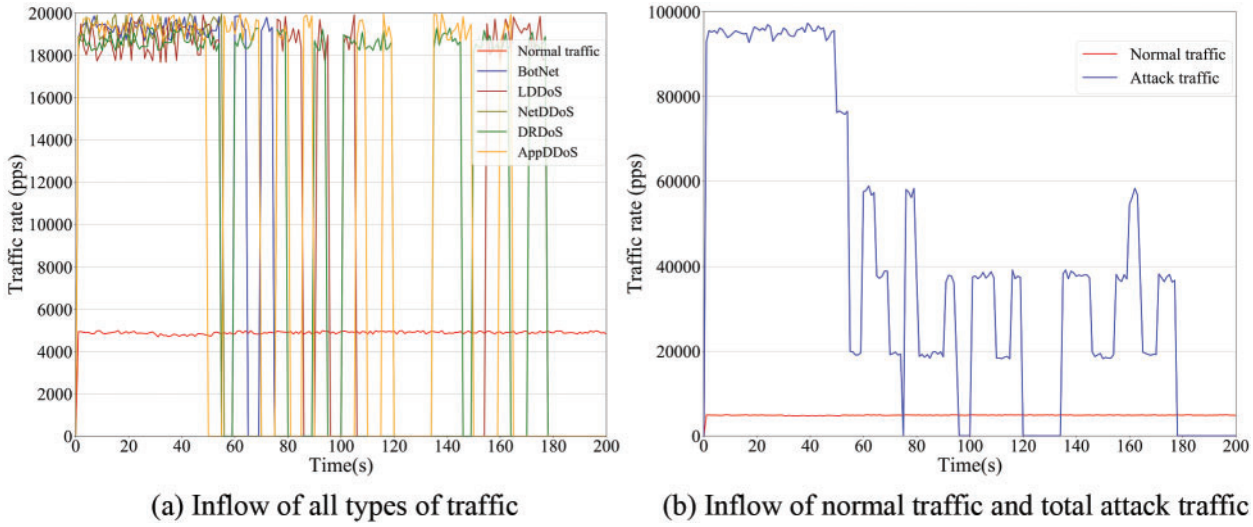


Figure 14: Comparison of reputation methods



(a) Inflow of all types of traffic (b) Inflow of normal traffic and total attack traffic

Figure 15: Traffic inflow at the resource owner

7 Conclusion

We have proposed a blockchain-empowered TBAC system with user reputation evaluation. We deployed the proposed TBAC system in Hyperledger Fabric and made the experimental evaluation and security analysis. The results demonstrate that the TBAC system can efficiently respond to access requests based on tokens. The user reputation evaluation module can accurately evaluate the reputation and restrict abnormal users through feedback. By comparing with other methods, we demonstrate the conservativeness and sensitivity of our proposed reputation model. Finally, we analyze the security of the system and show that the system in this paper is equally applicable in the face of malicious attacks by evaluating the performance of TBAC in the DDoS attack environment.

The TBAC system in this paper formulates clear interaction rules between user, resource, and blockchain, and proposes a method that combines user reputation evaluation with access control. In

future work, we will further extend the reputation feedback method and design incentives to reward high-reputation users to enhance user experience.

Acknowledgement: We would like to express our sincere gratitude to the reviewers for their insightful and constructive comments that helped us improve this paper's quality.

Funding Statement: This paper is supported by NSFC under Grant No. 62341102, and National Key R&D Program of China under Grant No. 2018YFA0701604.

Author Contributions: The manuscript was written entirely by the authors. All authors made an equal contribution to the development of the paper. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study can be obtained from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. Chopra, K. Gupta and A. Lambora, "Future internet: The internet of things-a literature review," in *2019 Int. Conf. on Machine Learning, Big Data, Cloud and Parallel Computing*, Faridabad, India, pp. 135–139, 2019.

[2] F. A. Alaba, M. Othman, I. A. T. Hashem and F. Alotaibi, "Internet of things security: A survey," *Journal of Network and Computer Applications*, vol. 88, pp. 10–28, 2017.

[3] S. Rana and M. Dheerendra, "An authenticated access control framework for digital right management system," *Multimedia Tools and Applications*, vol. 80, pp. 25255–25270, 2021.

[4] S. Ravidas, A. Lekidis, F. Paci and N. Zannone, "Access control in Internet-of-Things: A survey," *Journal of Network and Computer Applications*, vol. 144, pp. 79–101, 2019.

[5] X. S. Shen, D. Liu, C. Huang, L. Xue, H. Yin *et al.,* "Blockchain for transparent data management toward 6G," *Engineering*, vol. 8, pp. 74–85, 2022.

[6] C. Liao, X. Cuan, J. Cheng and S. Yuan, "Blockchain-based identity management and access control framework for open banking ecosystem," *Future Generation Computer Systems*, vol. 135, pp. 450–466, 2022.

[7] D. Chhikara, S. Rana, A. Mishra and D. Mishra, "Blockchain-driven authorized data access mechanism for digital healthcare," *Journal of Systems Architecture*, vol. 131, pp. 102714, 2022.

[8] S. Rana, M. Dheerendra and S. Mukhopadhyay, "Blockchain-based multimedia content distribution with the assured system update mechanism," *Multimedia Tools and Applications*, vol. 80, pp. 29423–29436, 2021.

[9] E. Bellini, Y. Iraqi and E. Damiani, "Blockchain-based distributed trust and reputation management systems: A survey," *IEEE Access*, vol. 8, pp. 21127–21151, 2020.

[10] G. Fortino, L. Fotia, F. Messina, D. Rosaci and G. Sarné, "Trust and reputation in the Internet of Things: State-of-the-art and research challenges," *IEEE Access*, vol. 8, pp. 60117–60125, 2020.

[11] D. D. Downs, J. R. Rub, K. C. Kung and C. S. Jordan, "Issues in discretionary access control," in *1985 IEEE Symp. on Security and Privacy*, Oakland, CA, USA, pp. 208, 1985.

[12] H. Zhu, K. Lü and R. Jin, "A practical mandatory access control model for XML databases," *Information Sciences*, vol. 179, no. 8, pp. 1116–1133, 2009.

[13] D. Ferraiolo, C. Janet and D. R. Kuhn, "Role-based access control (RBAC): Features and motivations," in *Proc. of 11th Annual Computer Security Application Conf.*, New Orleans, LA, pp. 241–248, 1995.

[14] V. C. Hu, D. R. Kuhn, D. F. Ferraiolo and J. Voas, "Attribute-based access control," *Computer*, vol. 48, no. 2, pp. 85–88, 2015.

[15] S. Gusmeroli, S. Piccione and D. Rotondi, "A capability-based security approach to manage access control in the internet of things," *Mathematical and Computer Modelling*, vol. 58, no. 5–6, pp. 1189–1205, 2013.
[16] G. Gan, E. Chen, Z. Zhou and Y. Zhu, "Token-based access control," *IEEE Access*, vol. 8, pp. 54189–54199, 2020.
[17] S. Rouhani and R. Deters, "Blockchain based access control systems: State of the art and challenges," in *Int. Conf. on Web Intelligence*, Thessaloniki, Greece, pp. 423–428, 2019.
[18] Z. Gao, L. Cao and X. Du, "Research progress of access control based on blockchain," *Chinese Journal of Network and Information Security*, vol. 7, no. 6, pp. 68–87, 2021 (In Chinese).
[19] S. Sun, S. Chen and R. Du, "Trusted and efficient cross-domain access control system based on blockchain," *Scientific Programming*, vol. 10, pp. 1–13, 2020.
[20] S. Rouhani, R. Belchior, R. S. Cruz and R. Deters, "Distributed attribute-based access control system using permissioned blockchain," *World Wide Web*, vol. 24, no. 5, pp. 1617–1644, 2021.
[21] H. Liu, D. Han and D. Li, "Fabric-IoT: A blockchain-based access control system in IoT," *IEEE Access*, vol. 8, pp. 18207–18218, 2020.
[22] R. Xu, Y. Chen, E. Blasch and G. Chen, "BlendCAC: A blockchain-enabled decentralized capability-based access control for IoTs," in *2018 IEEE Int. Conf. on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, Halifax, NS, Canada, pp. 1027–1034, 2018.
[23] Y. Chen, L. Tao, B. Liang, L. Sun, Y. Li *et al.,* "Capability- & blockchain-based fine-grained and flexible access control model," *IEEE Network*, pp. 1–8, 2023. https://doi.org/10.1109/MNET.127.2200414
[24] M. Ghafoorian, D. Abbasinezhad-Mood and H. Shakeri, "A thorough trust and reputation based RBAC model for secure data storage in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 4, pp. 778–788, 2018.
[25] M. Yang, "TDACS: An ABAC and trust-based dynamic access control scheme in hadoop," arXiv preprint arXiv:2011.07895, 2020.
[26] B. Gwak, J. H. Cho, D. Lee and H. Son, "TARAS: Trust-aware role-based access control system in public Internet-of-Things," in *2018 17th IEEE Int. Conf. on Trust, Security and Privacy in Computing and Communications/12th IEEE Int. Conf. on Big Data Science and Engineering (TrustCom/BigDataSE)*, New York, NY, USA, pp. 74–85, 2018.
[27] Z. Zhao and Y. Liu, "A blockchain based identity management system considering reputation," in *2019 2nd Int. Conf. on Information Systems and Computer Aided Education (ICISCAE)*, Dalian, China, pp. 32–36, 2019.
[28] G. D. Putra, V. Dedeoglu, S. S. Kanhere and R. Jurdak, "Trust management in decentralized IoT access control system," in *2020 IEEE Int. Conf. on Blockchain and Cryptocurrency (ICBC)*, Toronto, ON, Canada, pp. 1–9, 2020.
[29] G. D. Putra, V. Dedeoglu, S. S. Kanhere, R. Jurdak and A. Ignjatovic, "Trust-based blockchain authorization for IoT," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1646–1658, 2021.
[30] A. Dubey and V. Mishra, "Crowd review and attribute-based credit computation for an access control mechanism in cloud data centers," *International Journal of Computers and Applications*, vol. 45, no. 2, pp. 212–219, 2023.
[31] R. S. Sandhu and P. Samarati, "Access control: Principle and practice," *IEEE Communications Magazine*, vol. 32, no. 9, pp. 40–48, 1994.
[32] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, pp. 21260, 2008.
[33] H. Song, Z. Tu and Y. Qin, "Blockchain-based access control and behavior regulation system for IoT," *Sensors*, vol. 22, no. 21, pp. 8339, 2022.
[34] M. Li, H. Zhou and Y. Qin, "Two-stage intelligent model for detecting malicious DDoS behavior," *Sensors*, vol. 22, no. 7, pp. 2532, 2022.