electronics

MDPI

Article

Attribute and User Trust Score-Based Zero Trust Access Control Model in IoV

Jiuru Wang, Zhiyuan Wang, Jingcheng Song * and Hongyuan Cheng

School of Information Science and Engineering, Linyi University, Linyi 276000, China; wangjiuru@lyu.edu.cn (J.W.); 210854002012@lyu.edu.cn (Z.W.); chenghongyuan@lyu.edu.cn (H.C.)
* Correspondence: songjingcheng@lyu.edu.cn

Abstract: The Internet of Vehicles (IoV) is an innovative area of interest in modern mobility that is rapidly evolving while facing complex challenges. Traditional IoV networks are susceptible to intrusion threats, which can lead to data leakage and seizure of vehicle control by attackers, thereby endangering vehicle users' privacy and personal safety. An Attribute and User Trust Score-based Zero Trust Access Control Model (AU-ZTAC) is proposed, combining the zero-trust and attribute-based access control models to meet network protection requirements while achieving fine-grained dynamic access control and incorporating trust evaluation in the access control process to better reflect users' intent. Experimental results demonstrate the effectiveness and feasibility of trust assessment through the proposed model. A comparison with the classical schemes illustrates that AU-ZTAC allows for more flexible and fine-grained access control in complex access control environments while improving IoV security.

Keywords: zero trust; access control; trust score; Internet of Vehicles

Citation: Wang, J.; Wang, Z.; Song, J.; Cheng, H. Attribute and User Trust Score-Based Zero Trust Access Control Model in IoV. *Electronics* **2023**, *12*, 4825. https://doi.org/10.3390/electronics12234825

Academic Editor: Dongkyun Kim

Received: 20 October 2023
Revised: 25 November 2023
Accepted: 27 November 2023
Published: 29 November 2023

1. Introduction

In recent years, the growing requirement for autonomous driving and the rapid development of the Internet of Things (IoT) has driven great global attention to the Internet of Vehicles (IoV). In broad terms, IoV represents the fusion of Vehicular Ad Hoc Networks (VANET) with IoT [1]. In the present day, IoT technology empowers connected vehicles to link with networks and access real-time traffic data, navigation assistance, and other driving-related functionalities. According to Gartner, 5G IoT is poised to be the leading communication technology for connected automobiles. Gartner predicts that by 2030 a substantial portion of the 5G IoT market will be dedicated to the automotive sector, with connected cars projected to encompass approximately 53% of the total 5G IoT endpoints [2].

Security and trust have garnered widespread attention in traditional IoV architectures due to susceptibility to malicious attacks, malfunctions, or erroneous data, which can lead to serious security issues [3]. In traditional autonomous vehicle network architecture there is a model of mutual trust between the various components and sensors of the vehicle, which means that they assume by default that all other systems and sensors are trustworthy. However, this trust model is vulnerable to malicious activities and can lead to serious security issues. First, data credibility is a core issue, as autonomous vehicles rely on sensors and data to sense the surrounding environment [4]. A second important issue data is data integrity. An attacker may tamper with sensor data to mislead the autopilot system, e.g., by changing traffic signs or vehicle position information [5]. The security concerns related to the vehicle's internal software are equally severe. If hackers manage to infiltrate the system and gain control of the vehicle, the consequences could be catastrophic, potentially resulting in severe accidents or criminal activities.

In addition to these, there are a number of access control problems in autonomous vehicle networks that can reduce safety and reliability. Autonomous driving systems typically comprise multiple components and subsystems within various trust boundaries [6]. The

intricate nature of autonomous vehicle networks introduces complex challenges related to access control, posing significant implications for the overall security and functioning of these systems. The problem with access control is that if one component is attacked or vulnerable, an attacker may be able to cross the trust boundary and affect other components. This can lead to system-wide security breaches. In the realm of autonomous vehicles, the need for robust access control extends beyond external threats to encompass internal risks as well. Threats may emanate from components or users within the vehicle itself that act maliciously or engage in improper operations. This internal dimension adds another layer of complexity to the access control challenge, requiring comprehensive solutions to mitigate potential risks [7]. Autonomous vehicles must respond to changing road and traffic conditions in real-time. Therefore, access control must be flexible, allowing legitimate and necessary communication and data sharing while discouraging malicious behaviour. Autonomous vehicles collect large amounts of sensor data, including camera images and sensor readings. Access control must protect these sensitive data against unauthorised access or data leakage [8]. Various components and subsystems on a vehicle require authentication and authorisation to ensure that only legitimate users or components can access specific functions and data. If authentication or authorisation is inadequate, a malicious user may abuse the system's privileges.

Zero-trust architecture is adopted for solving above problems, which centers on trusting no component or data even if it is from internal system entities. Zero-trust architectures represent a paradigm shift in cybersecurity; by adopting a zero-trust mindset, organizations can fortify their autonomous vehicle networks against evolving security threats. Zero-trust architectures can improve the security of network architectures for autonomous vehicles in several ways. First, data validation is crucial, and all sensor data must be verified to ensure its origin and integrity using techniques such as digital signatures, encryption, and data integrity to check the data and protect it against tampering. Second, access control must be carried out according to the principle of least privilege, allowing communication between components only when needed in order to reduce the potential attack surface. In addition, multi-layered security measures, including network isolation, intrusion detection systems, real-time monitoring, and an emergency response plan, can provide a rapid response if a system suffers a security incident.

In this paper, an Attribute and User Trust Score-based Zero Trust Access Control Model (AU-ZTAC) is proposed in which the zero-trust architecture is adopted for better security and Attribute Based Access Control (ABAC) is used for fine-grained access. Trust evaluation is adopted to reflect better users' actual behaviours and intentions. Trust scores are calculated through the Fuzzy Analytic Hierarchy Process (FAHP). This model provides fine-grained access control while enhancing the security and accuracy of access control by integrating attributes and user trust scores.

In AU-ZTAC, users can access a resource when the attribute meets the requirements of the access control policy and the trust score is higher than the threshold for accessing the resources. In this way, the purpose of protecting sensitive data and resources is achieved and the current network security challenges are effectively overcome.

The main innovations of this paper are listed as follows:

(1) Attribute and user trust score-based zero trust access control model. A novel access control model that combines zero-trust architecture and trust evaluation on the basis of ABAC is proposed to achieve more accurate and reliable access control. At the same time, the proposed model can effectively deal with the defects exposed by traditional autonomous vehicle network architectures.

(2) Fine-grained access control. By adding trust evaluation to ABAC, the model achieves fine-grained access control; the subject attributes must match the policy and satisfy the predefined trust score threshold, providing more precise and personalized privilege assignment.

(3) Zero-trust architecture. The model is based on the zero-trust architecture concept, and achieves fine-grained dynamic access control of the zero-trust architecture by combin-

ing attribute-based access control and calculating trust scores through FAHP. An effective response to external intrusion and internal attacks improves resource security in the current network environment.

The remainder of this paper is structured as follows: Section 2 briefly reviews related work; Section 3 introduces the materials and methods used in AU-ZTAC; Section 4 describes the method for evaluating trust scores; Section 5 experimentally verifies the feasibility of trust scores and presents a comparison with other classical models; finally, the paper ends with a discussion and conclusions in Section 6.

2. Related Work

This section presents related research work, including access control schemes using role-based access control, attribute-based access control, new access control policies and methods, and dynamic access control.

The Role-Based Access Control (RBAC) model assigns and manages privileges based on user roles, each with a set of privileges, with users assuming one or more roles. Based on RBAC, Wang et al. [9] proposed an access control model based on role-based access control. They simultaneously used blockchain and user creditworthiness to store the access control policy on a blockchain in the form of smart contracts. To address the problem of no access control model being able to enforce the permissions of safety officers, Habib et al. [10] proposed a novel Security and Privacy Based Access Control (SPBAC) model for vehicle internet connections that allows safety officers to access the combined permissions and roles, rather than just the roles, of information in vehicles belonging to the same fleet. Chatterjee et al. [11] proposed a new dynamic role-based access control framework for decentralized applications (dApps). The framework allows management access control on dApps, where the dApp is fully decoupled from the business application and seamlessly integrated with any dApp. Se-Ra Oh et al. [12] introduced an interoperable access control framework that is based on OAuth 2.0 and roles, enabling client-specific domains in heterogeneous IoT platforms to utilize the IAT for access and permitting authenticating roles to enter the proposed framework and subsequently access valuable resources in the domains. Abdul et al. [13] designed an access control mechanism based on RBAC and calculated the trust level based on the user's uncertain behaviour. This mechanism mitigates malicious operations caused by authenticated users. However, RBAC is generally less dynamic in the assignment of roles. When a role has been assigned, a user's privileges usually remain unchanged for an extended period. RBAC is better suited to more straightforward privilege management needs, especially in large organizations where the assignment of privileges is relatively static and role-based.

ABAC is a flexible access control policy that makes access control decisions based on multiple attributes, such as users, resources, and environments. Based on ABAC, Belchior et al. [14] proposed a Self-Sovereign Identity-Based Access Control (SSIBAC). This model is used for cross-organizational identity management and provides decentralized authentication followed by centralized authorization using traditional access control models and blockchain technology. To address the problem of access control in the Internet of Industrial Vehicles (IIoV) ecosystem, Gupta et al. [15] proposed a formal attribute-based access control system, a model that introduces the concept of groups, which are assigned to various intelligent entities based on different attributes. Bhatt et al. [16] developed a formal ABAC model for Amazon Web Services (AWS) IoT by building and extending the previously developed AWS IoT access control model, which combines its existing functionality with introducing new attributes and attribute-based policies for IoT entities to enable expressive access control in AWS IoT. Challagidad et al. [17] proposed an efficient multi-privilege access control using an attribute-based encryption scheme. The proposed scheme, which consists of a Role Hierarchy Algorithm (RHA) and Hierarchical Access Structure (HAS), protects user data to ensure privacy, multi-privilege access control, and fine-grained access to stored data. Ezhil et al. [18] proposed a new auditable attribute-based encryption scheme for data-sharing systems that combines the benefits of blockchain

technology with attribute-based access control by using blockchain to provide secure attribute-based data sharing and integrity auditing while providing compensation to data owners if their data integrity is lost. García-Teodoro et al. [19] introduced a novel implementation of attribute-based access control called Dynamic Access Control Based on Security Attributes, in which the system actively evaluates the security profiles of subjects in a communication environment, then grants, restricts, or denies access to network services and resources over time. ABAC is better suited to access control needs that require fine-grained, dynamic, and flexible access control that can make decisions based on multiple attributes while adapting to more complex contexts. ABAC allows organizations to control access to resources based on complex policies and conditions, making it highly superior in dynamic and complex environments.

Researchers have proposed new access control strategies and methods in other aspects of access control as well. Access control strategies and methods are critical to protecting information assets and system security. They help organizations to achieve security and compliance by restricting and managing access rights and providing flexible access control capabilities. These policies and methods involve different access control aspects. For example, DeCusatis et al. [20] proposed implementing a zero-trust cloud network with transport access control and first-packet authentication, achieving a zero-trust network through first-packet-based authentication. However, they only studied the target user behaviour measurement metrics and measurement algorithms, and did not present a feasible architecture for implementing a zero-trust system. Vanickis et al. [21] focused on implementing access control policies on Zero-Trust Networks (ZTN). The authors devised a mechanism for mapping these rules to a specific firewall syntax and installing these rules on the firewall. However, their it did not include development language tools and was not integrated into existing Professional Dynamic Programming (PDP) systems, meaning that it cannot guarantee the stability of the runtime mechanism. Mandal et al. [22] proposed a novel access control policy based on zero-trust networks to substantiate MAC spoofing attacks in the software-defined networking (SDN) cloud computing paradigm by restricting incoming network traffic. However, under the security threat of advanced attackers it is not possible to guarantee the reduction of the threshold and optimal security of cloud resources, and this approach does not address the time-consuming nature of analyzing traffic and weeding out spoofed users. In modern complex network environments, relying solely on static policies and rules may not adequately address security needs. Therefore, researchers have used dynamic access control to cope with situations where permissions need to be changed frequently or dynamically adjusted based on context. For example, Guo et al. [23] proposed a dynamic trust evaluation algorithm based on zero-trust architecture research to protect the sensitive information resources of the system by dynamic control of access authorization through multi-source attribute acquisition and attribute encryption. However, they did not detail the access control method used to achieve zero-trust dynamic access control. Yao et al. [24] proposed a dynamic fine-grained access control and authorization system that calculates user trust based on the degree of deviation from the user's current and historical behaviour. However, due to the TBAC model, tasks and roles cannot be separated, and passive access control and role hierarchy are not supported. Lin et al. [25] proposed an adaptive data lifecycle-oriented access control method called AHCAC for implementation in hybrid cloud environments. It achieves fine-grained dynamic access control of data in the hybrid cloud environment through a key attribute-based policy description and a policy distance hierarchical clustering algorithm. However, this model mainly applies to cloud environments, and cannot effectively deal with the problems exposed by traditional boundary-based network architectures.

In response to the above-mentioned problems, in this paper we propose an attribute and user trust score-based zero trust access control model that establishes access decisions based on subject, resource, and environment attributes using ABAC. The user's trust level is quantified by the access control through trust evaluation. This can solve the problems of lack of dynamism and insufficient fine-grainedness in role-based access control

modelling schemes. The FAHP comprehensively considers the user's credibility based on multiple factors, and is applied in ABAC to reflect the user's actual intention more comprehensively. In addition, the model employs a zero-trust architecture along with the mechanisms of ABAC. The zero-trust architecture does not consider any user or device to be inherently trustworthy, requiring continuous authentication and authorization of each access request. ABAC enables fine-grained control of access requests concerning specific attribute conditions. By combining ABAC with a zero-trust architecture, the proposed model can simultaneously achieve fine-grained access control and counteract network insider attacks. A comparison of AU-ZTAC with other approaches in terms of applicability, dynamism, flexibility, fine-grained control, trust evaluation, and simplicity is shown in Table 1.

Table 1. Comparison of AU-ZTAC with other approaches.

Model/Method	Applicability	Dynamism	Flexibility	Fine-Grained Control	Trust Evaluation	Simplicity
AU-ZTAC	High	High	High	Yes	Yes	Medium
RBAC + Blockchain + Creditworthiness	Low	Low	Low	No	Yes	Low
SPBAC (Security and Privacy Based AC)	Medium	Medium	Medium	Yes	No	Low
Dynamic RBAC	Medium	Medium	Low	No	No	Medium
SSIBAC (Self-Sovereign Identity AC)	Medium-High	High	Medium	Yes	No	Low
ABAC (Attribute-Based AC)	Medium	High	High	Yes	No	High
Dynamic ABAC	Medium	High	High	Yes	No	Medium
Multi-Privilege ABAC	Medium	High	Medium	Yes	No	Low
Auditable ABAC	High	High	Medium	Yes	No	Medium
Zero Trust Cloud Network	Low	High	High	Yes	No	Medium
Dynamic Fine-Grained AC	Medium	High	Medium	Yes	No	Medium

3. Materials and Methods

3.1. Zero-Trust Architecture

With the increase in the size and complexity of network systems, the traditional perimeter-based network security model for autonomous vehicle drivers can no longer meet the security requirements of rapidly evolving network systems. To solve this problem, zero trust was first proposed in 2010 [26] and applied through Google's BeyondCorp project [27]. The innovation of the zero-trust architecture is that it abandons the traditional perimeter-based security model by deploying all applications on the public network, meaning that it no longer divides the network into internal and external. Zero-trust architectures accept that threats are everywhere in the network, and assume that the environment is perpetually insecure. Authorization no longer relies on network location. All devices, users, and network traffic must be authenticated and authorized according to a dynamic security policy, and continuous authentication is necessary for authorization even within the network. This approach enables secure access from any location using any device, and can fulfill the current demands of telecommuting.

The essence of zero trust is identity-centered dynamic access control [28]. It fundamentally mistrusts devices, users, and network traffic within and outside the network. In the zero-trust architecture, instead of relying on static rules to grant privileges, the network continuously authenticates identities to assess their trust level in real time based on user behaviour when users request access to resources, while dynamically assigning privileges based on a trust score. The core components of the zero-trust architecture are shown in Figure 1.

The supporting system of the zero-trust architecture is referred to as the Control Plane, while the remaining components constitute the Data Plane. Logical components

communicate through a dedicated Control Plane, whereas application data are transmitted over the Data Plane. The Data Plane is directed and configured by the Control Plane, and requests to access protected resources undergo initial processing in the Control Plane. This processing includes authentication and authorization of devices and users, with fine-grained access control policies implemented at this layer as well. In scenarios where a user is situated in an untrusted area and seeks to access resources within a trusted domain, the initial step involves authentication before allowing entry into the system. Subsequently, the user secures permissions by interacting with the Policy Enforcement Point (PEP) and Policy Decision Point (PDP).

The policy engine (PE) is the critical component that decides whether or not to grant a subject access to a resource. It makes access decisions based on security policy and input from external sources (e.g., IP blacklists, threat intelligence services) and outputs the results of access authorization or denial.

The policy administrator (PA) establishes and breaks the communication path between the subject and the resource by giving commands to the Policy Enforcement Point PEP. Suppose the policy engine agrees to grant access to a resource. In this case, the PEP generates an authentication token or credentials for accessing the resource. In the case of the policy engine denying the user access to the resource, the policy manager signals the PEP to cut off the communication path between the user and the resource.

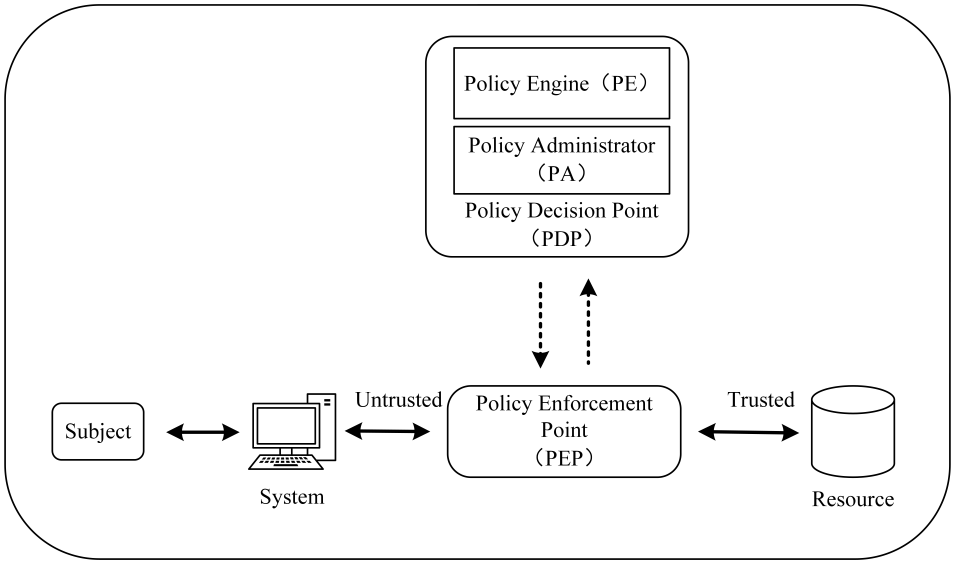

Figure 1. Zero-trust architecture component diagram.

The Policy Enforcement Point (PEP) enables, monitors, and ultimately terminates connections between access subjects and resources. It communicates with the policy administrator to forward requests or receive policy updates.

In addition to the policy engine, policy administrator, and policy enforcement points, certain data sources may provide inputs and decision rules for the policy engine to use in making access decisions. These data sources can include threat intelligence sources, data access policies, identity management systems, etc. Together, these components and data sources form a crucial part of the zero-trust architecture, ensuring that only authorized subjects can access resources while providing flexible access control capabilities.

3.2. Attribute-Based Access Control

Access control is widely used as an essential means of protecting data resources [29]. It includes several major models, such as Discretionary Access Control (DAC), Mandatory Access Control (MAC), and RBAC, all of which achieve authorisation and control of access to objects by different subjects [30]. However, these models have limitations in certain areas. The propagation of privileges in the DAC mechanism is a security risk, and the

inefficiency of today's increasingly complex systems is becoming apparent. In the MAC mechanism, the subject must comply with the system's access control policy, which the system administrator formulates to realise multi-level access control by dividing the subject and the object into different security levels. However, this approach makes it very difficult to change privileges. This lack of flexibility restricts its application to a specific system, making it non-universal. Instead of being applied directly to the user's rights RBAC is applied to each user to assign access rights before the user is assigned roles. These roles represent the user's different access rights. However, when the RBAC model is faced with more complex environments, such as when there is an increase in the number of users, it needs to assigned more different roles, leading to user roles and the relationship between roles and privileges becoming very intricate, complex, and difficult for administrators to manage. By introducing attributes between users and privileges as a judgment condition, ABAC can effectively avoid the problem of too many roles that arises in RBAC. Users are granted privileges based on their different attributes, allowing for fine-grained privilege distribution and support for larger-scale complex systems.

ABAC is a logical access control method [31]. The components and access control flow are shown in Figure 2. Instead of granting privileges based on user identity, ABAC uses entity attributes, introducing these attributes as judgment conditions between users and privileges. The authorisation of ABAC seeks to match the combination of a visitor's attributes with the access control policy at the time of the user's access request and then grant privileges by assessing whether the subject and object attributes, type of operation, and relevant environmental conditions comply with the access control policy [32]. The relationship between the attributes in the ABAC model is not static, and can be adapted according to actual needs. When the access control policy is changed, only the value of the attribute needs to be changed, without any need to change the underlying rules of the policy. Hence, ABAC is a flexible access control model that can achieve fine-grained dynamic access control.

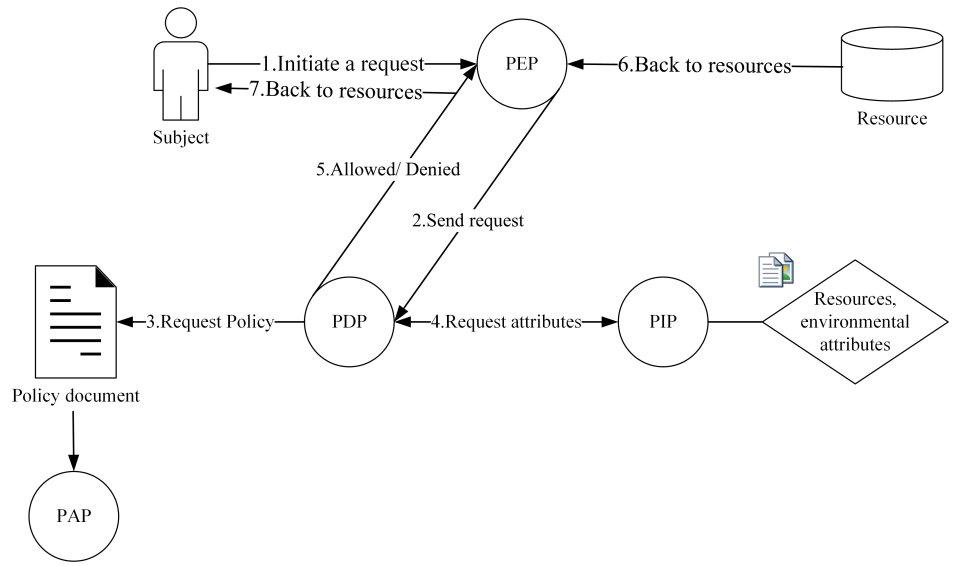

Figure 2. Attribute-based access control model diagram.

The ABAC process involves a systematic approach to managing access to resources based on various attributes. In this method, access control decisions are made by evaluating the attributes associated with the subject, action, resource, and environment. The ABAC process begins with the identification of the subject seeking access, which can be a user, system, or application. Subsequently, the specific action or operation intended by the subject is considered along with the attributes associated with the targeted resource and the contextual environment in which the access request occurs. These attributes collectively

contribute to the formulation of access control policies that determine whether the requested action should be permitted or denied. The specific access control process is as follows:

(1) The user attempts to access a resource by sending an original request, which is intercepted by the Policy Enforcement Point (PEP).

(2) The PEP forwards the access request to the Policy Decision Point (PDP). The PDP evaluates the request and requests policies from the policy document.

(3) Policies are stored in the policy document and maintained by the Policy Administration Point (PAP).

(4) The PDP receives the evaluation results of the access request and forwards them to the PEP.

(5) Based on the received information, the PEP either grants or denies the user's access to the resource.

The ABAC model involves several components, each of which plays a specific role. The following is a brief description of the components of the ABAC model.

Subject: the subject represents the entity that accesses the resource, such as a user, device, application, etc. The subject typically possesses related attributes, such as roles, department, location, etc., which can be assessed when making access decisions.

Resource: a resource is a protected object, such as a file, database, network service, etc. Resources usually have related attributes, such as a sensitivity level, owner, etc., that can be referred to when making access decisions.

Environment: the environment refers to the context in which an access request occurs, such as the time, location, network connectivity status, etc. Environmental attributes can affect access decisions, such as allowing access during a specific period or restricting access privileges in a particular network environment.

Attribute: attributes are characteristics or features that describes a subject, resource, or environment. Attributes can be any information related to users, roles, resources, environments, etc. For example, the position of a user, the permission set of a role, the resource's sensitivity level, the network status of an environment, etc., can all be used as attributes.

Policy: the policy defines the access control rules and conditions that specify which attribute values allow access to which resources and under what circumstances access is to be permitted or denied.

Policy Decision Point (PDP): the PDP is the core component responsible for making access decisions. It receives access requests, evaluates them according to predefined policies, and decides whether to allow access based on attribute matching and logical rules.

Policy Information Point (PIP): the PIP is responsible for obtaining attribute values from various data sources to meet the needs of access decisions. It can interact with the user attribute store, resource attribute store, identity provider, etc., to obtain necessary attribute information.

Policy Enforcement Point (PEP): the PEP is an enforcement point located on the access path to a resource that enforces access control based on the PDP's decision-making. The PEP, which may be part of a network device, an application, or an operating system, is responsible for interrupting or permitting access requests to ensure that only authorised subjects can access the resource.

3.3. AU-ZTAC Design Objectives

AU-ZTAC aims to ensure the security and reliability of resource access. It achieves fine-grained access control and dynamic authorization by integrating zero trust and ABAC. This allows users to undergo dual verification through attribute matching and trust evaluation when accessing resources. The design objectives can be summarized as follows:

- Fine-grained access control: this refers to the precise control of resources or data at a granular level within a system. It ensures that users or roles can only access the minimum scope or specific parts required, thereby enhancing security and safeguarding sensitive information. AU-ZTAC enables precise control over which attributes can

access which resources by defining attribute rules and policies in order to achieve fine-grained access control. Real-time access decisions are made based on user attributes and behaviours.

- Dynamic authorization: this refers to granting or denying access privileges to users or principals in real time based on contextual information and policy evaluation. It incorporates the concept of trust assessment, where access privileges to resources are determined based on the user's attributes and trust score. Access privileges can be adjusted accordingly when there are changes in the user's attributes or trust score. For instance, if a user's trust score decreases or if specific attributes change, the system can dynamically revoke or reduce access privileges in response. Similarly, if a user's trust score increases or particular conditions are satisfied, the system can dynamically grant higher-level access privileges. By combining user trust scores with real-time access decisions, AU-ZTAC enables a dynamic authorization mechanism.
- Security and reliability: to ensure the utmost security and reliability while accessing resources, features such as continuous trust assessment, continuous authentication, and dynamic access control are used to build a line of security defence that reduces unauthorized access and mitigates security risks.

3.4. AU-ZTAC Method Overview

Figure 3 provides an in-depth overview of the AU-ZTAC architecture, which is implemented in two key phases: (1) integration of ABAC within a zero-trust framework, facilitating fine-grained dynamic access control; and (2) introduction of trust score calculation and evaluation during the access control process.

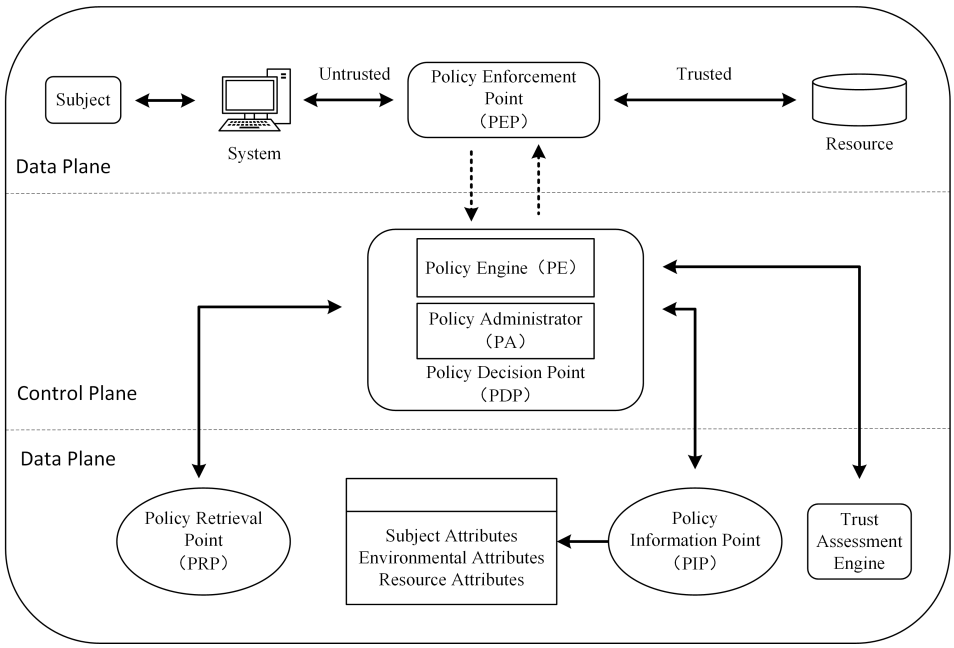

Figure 3. Architectural diagram of attribute and user trust score-based zero trust access control model.

The control plane is the supporting system of the zero-trust architecture, while the other parts constitute the data plane. Communication among logical components occurs through a dedicated control plane, while the data plane handles the transmission of application data. The access control process relies on ABAC, allowing the definition of flexible access control policies based on diverse attribute criteria. Its primary strength lies in its ability to formulate access control conditions by combining a range of attribute information, thereby adapting flexibly to various resource access scenarios. The incorporation of trust evaluation in the access control process provides a more comprehensive reflection of user intent, with the trust score calculated through FAHP.

In scenarios where a user in a non-trusted zone seeks to access resources in a trusted zone, the process involves initial authentication for system entry. Subsequently, access control and trust assessment determine the permissions to be granted. This includes matching user attributes with access control policies, calculating the trust score, and ensuring that it meets the preset threshold for the resource.

The stringent access criteria require users to align their attributes with the access control policy and achieve a trust score meeting the resource-specific threshold. Successful passage through the Policy Decision/Execution Point (PDP/PEP) grants users access privileges to trusted zone resources, encompassing systems, data, applications, and other critical assets. Access privileges are contingent on satisfaction of the access control policy conditions and attainment of a trust score meeting the predefined threshold for the resource.

In today's threat landscape, the integration of a zero-trust architecture with ABAC and trust evaluation ensures that only authorized and trust-evaluated users can access resources within trusted zones. This mitigates unauthorized access, thereby enhancing overall security. The zero-trust architecture, with its continuous trust assessment, ongoing identity authentication, least-privilege allocation, and dynamic access control, establishes a cutting-edge network security framework. Leveraging ABAC and dynamic trust scoring enables real-time access decisions based on user attributes and behavior, ensuring that only authorized and trusted users can access sensitive resources. This fine-grained access control offers flexibility and security, resulting in a robust security defence line for organizations.

3.5. AU-ZTAC Access Control Process

AU-ZTAC aims to allow a user in an untrusted zone to gain access to resources in a trusted zone through continuous authentication and authorization. Figure 4 depicts the detailed authorization flow.

(1) When a user accesses a resource, the user first sends an original request. The trust evaluation engine evaluates the user's trust score according to the user's behavioural characteristics when the user enters the system. The fifth section detail the evaluation method.

(2) The request is intercepted at the PEP and forwarded to the PDP.

(3) The PDP evaluates the access request and requests the access policy from the policy file.

(4) The various user attributes obtained in the PIP are matched with the access control policies in the policy file.

(5) When the attribute conforms to the access control policy, the trust score is compared with the predefined threshold of the resource.

(6) The results are sent to the PEP to determine whether the user can access the resources in the trusted area.

There is no difference between steps (1)–(4) and the steps of ABAC. In ABAC, the PEP can decide whether to grant access to the resource based on the result after matching the attribute and the access control policy. However, in AU-ZTAC, judgment based on trust score is added. After matching the policies and attributes, the access control process does not immediately transmit the result to the PEP; first, it compares the trust score against a preset threshold for the resource in question. Only when the attributes carried by the user match the access control policy and the trust score exceeds the threshold is the permission granted to the user. If either of these conditions does not meet the requirements, the PDP denies the user's request to access the resource.

The implementation process combines ABAC with zero trust. It utilizes trust evaluation to resolve the lack of trustworthiness in authorization and the inability of ABAC to fully capture user intentions. Furthermore, resource thresholds can be dynamically adjusted based on the circumstances. AU-ZTAC enhances the security and the flexibility of authorization, enabling more flexible and fine-grained access control.

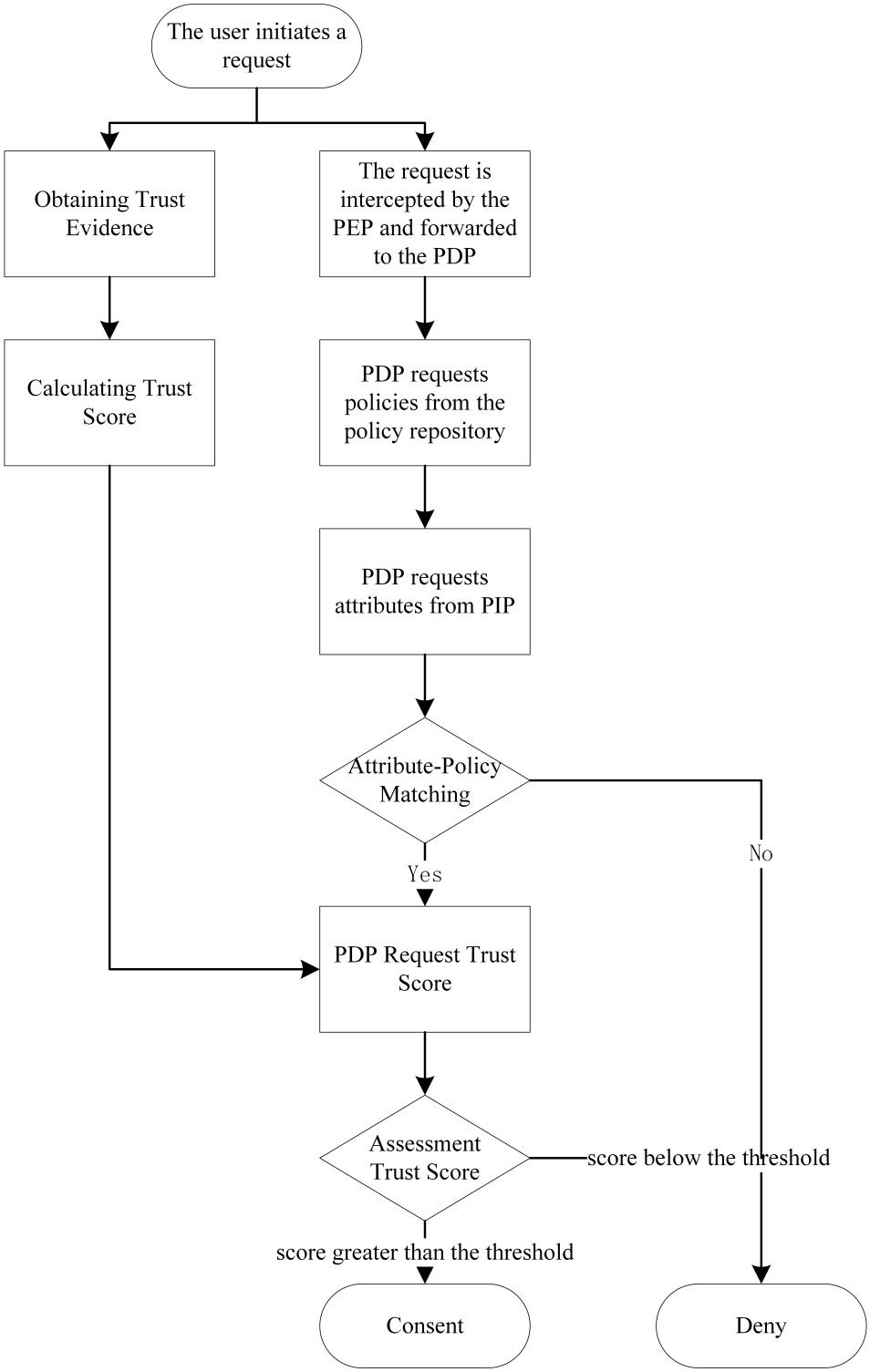

Figure 4. Flowchart of attribute and user trust score-based zero trust access control model.

4. User Trust Score Evaluation Method

The evaluation of user trust scores is an essential part of AU-ZTAC. Trust evaluation is a useful means to measure the reliability of an object's behavior [33]. FAHP is adopted for the evaluation of user trust scores. This method first divides user behaviour into n characteristics, then divides each characteristic into multiple evidence types to refine the fuzzy and uncertain user behaviour trust value evaluation problem into a simple, clear, and weighted summation problem of trust evidence.

The evaluation of trust based on user behaviour is fuzzy and difficult to calculate directly. It needs to first be divided into several characteristics, then these characteristics need to be subdivided into associated trust evidence. Figure 5 presents the division of characteristics and trust evidence. Initial user behaviour evidence can be obtained through network traffic monitoring tools, specialized data collection tools, or custom-developed tools based on specific requirements. This trust evidence are represented as $BE = (a_{ij})_{nm}$, where m is the maximum number of items in trust evidence. When the number of items for a particular piece of evidence is insufficient, it needs to be padded with zeros; n is the number of characteristics into which user behaviour trust is divided. After obtaining the user behaviour evidence, it is necessary to normalize the evidence into a dimensionless value that increases in the interval [0, 1], which can facilitate the calculation of the trust score. The normalized trust evidence is expressed as $E = (e_{ij})_{nm}$.

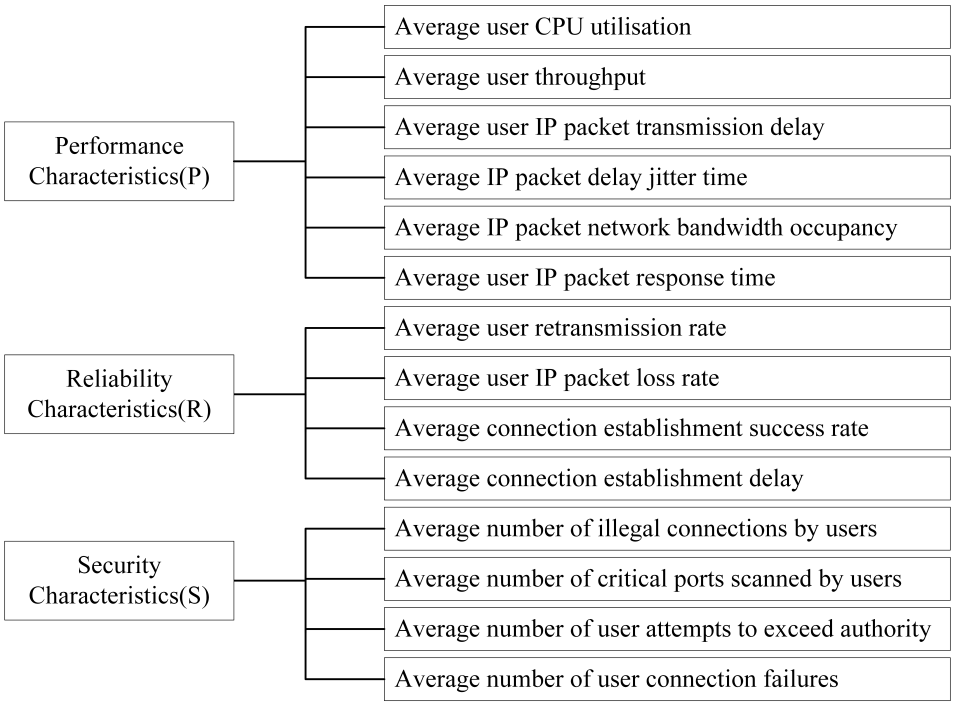

Figure 5. Evidence of user trust.

The judgment matrix measures the relationship between the evidence in the BE. To obtain the initial judgment matrix, the m pieces of trust evidence $E = (e_1, e_2, ..., e_m)$ need to be compared pairwise according to their relative importance. This process results in the initial judgment matrix $EQ = (eq_{ij})_{mm}$, where

$$eq_{ij} = \begin{cases} 0, & e_i < e_j \\ 0.5, & e_i = e_j \\ 1, & e_i > e_j. \end{cases} \tag{1}$$

The initial judgment matrix is then transformed into a fuzzy consistency matrix $Q = (q_{ij})_{mm}$, where q_{ij} and q_i are computed as

$$q_{ij} = \frac{q_i - q_j}{2m} + 0.5; q_i = \sum_{k=1}^{m} q_{ik}. \tag{2}$$

Next, we calculate the weight vector of m evidence of an attribute $W = (w_1, w_2, ..., w_m)^T$, where

$$w_i = \frac{\sum_{k=1}^{m} q_{ik} - 0.5}{m(m-1)/2}. \tag{3}$$

The evaluation value matrix of user behavioural characteristics derives from the evidence matrix $E = (e_{ij})_{nm}$ and the weight matrix $W = (w_{ij})_{mn}$. The value on the diagonal of the matrix obtained from $E \times W^T$ is the evaluation value matrix of the characteristics $F = (f_1, f_2, ..., f_n)$. The current user's behavioural trust score is

$$T = 1 - F \times W_f^T = 1 - \sum_{i=1}^{n} f_i w_i, \tag{4}$$

where $W_f = (w_{f1}, w_{f2}, ..., w_{fn})$ is the weight of the user behavior characteristics.

User behavioural trust occurs periodically, and access subjects usually have multiple access behaviours. A user's historical trustworthiness should influence their current trustworthiness; hence, we introduce a time decay factor γ, with the value of γ falling between 0 and 1. The final trustworthiness of the user in the current period T_l is calculated based on the trustworthiness from the previous period T_{l-1} and the currently computed trustworthiness T:

$$T_l = \gamma T_{l-1} + (1 + \gamma)^T. \tag{5}$$

5. Experimental Design and Analysis of Results

5.1. Experimental Configuration

The experiment and results analysis demonstrates the calculation of user trust scores. During user access, the traffic monitoring program obtains the user trust evidence value, normalises it, and calculates it. The experiment was run on a VMware virtual machine installed on a physical machine, using PyCharm as the Integrated Development Environment. Tables 2 and 3 show the configuration of the physical and virtual machines, respectively.

Table 2. Physical machine configuration.

System Configuration	Configuration Information
Operating System	Windows 11
Version	22H2
RAM	16 GB
Processor	R7 5800H
SSD	512 GB

Table 3. Virtual machine configuration.

System Configuration	Configuration Information
Operating System	Linux(L)
Version	CentOS 7 64-bit
RAM	1 GB
Number of Processors	1
Number of Cores per Processor	1
Hard Disk	20 GB

5.2. Calculation Procedure

Because it is challenging to measure the degree of trust during the actual access resource interaction process using the calculated trust score, the user trust level is set to five levels according to the definition of trust level in [34] and the interval of each interval is set to (0.05, 0.25, 0.60, 0.80). Each interval of the trust value corresponds to a trust level.

The traffic monitoring program measures user trust evidence values during the system access process. The trust evidence value has various forms of expression, such as percentage, specific numerical value, etc. It is necessary to standardize the trust evidence into dimensionless values that increase within the interval [0, 1] in order to facilitate the calculation of trust scores. The standardization method is as follows:

The average user CPU utilization and average IP packet network bandwidth occupancy are measured by the performance characteristic P. The average user retransmission

rate, average user IP packet loss rate, and average connection establishment success rate in the reliability characteristic S are percentage expressions, which are already in the interval [0, 1], and the normalization method is:

$$e_{ij} = a_{ij}. \tag{6}$$

Other specific numerical manifestations of trust evidence normalized to be within the interval [0, 1] are

$$e_{ij} = \frac{a_{ij} - (a_{ij})_{min}}{(a_{ij})_{max} - (a_{ij})_{min}}. \tag{7}$$

The normalized evidence of trust is as follows:
$P = [p_1, p_2, p_3, p_4, p_5, p_6] = [0.979, 0.668, 0.622, 0.539, 0.010, 0.176]$
$R = [r_1, r_2, r_3, r_4] = [0.290, 0.490, 0.670, 0.661]$
$S = [s_1, s_2, s_3, s_4] = [0.035, 0.179, 0.028, 0.434]$

The calculation procedure is illustrated by the reliability characteristic R. The testing experience with the existing networks shows that the importance of trust evidence for the reliability characteristic is $r_1 = r_2 < r_3 = r_4$ and the initial judgment matrix EQ is obtained from Equation (1):

$$EQ = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 1 & 1 & 0.5 & 0.5 \\ 1 & 1 & 0.5 & 0.5 \end{bmatrix}.$$

The initial judgment matrix EQ is transformed into a fuzzy consistency matrix Q using Equation (2):

$$Q = \begin{bmatrix} 0.5 & 0.5 & 0.25 & 0.25 \\ 0.5 & 0.5 & 0.25 & 0.25 \\ 0.75 & 0.75 & 0.5 & 0.5 \\ 0.75 & 0.75 & 0.5 & 0.5 \end{bmatrix}.$$

The weight vector $W_r = [0.167, 0.167, 0.33, 0.33]^T$ results from Equation (3) for the four pieces of evidence for the reliability attribute R. Similarly, the importance of the functional attribute P is $p_1 = p_5 = p_6 > p_2 > p_3 = p_4$ and the weight vector is $W_p = [0.217, 0.15, 0.1, 0.1, 0.217, 0.217]^T$, while the importance of the security attribute S is $s_2 = s_3 > s_1 > s_4$ and the weight vector is $W_s = [0.208, 0.333, 0.333, 0.125]^T$.

According to the values on the diagonal of the matrix obtained by $E \times W^T$, the matrix of evaluation values of the characteristics is $F = [0.469, 0.574, 0.131]$, the importance of the user's behavioural characteristics is $P < R < S$, and, following the same calculation process as in the above example, the weight of the behavioural characteristics is obtained as $W_f = [0.167, 0.333, 0.5]$.

Finally, Equation (4) yields the user's credit score as $T = 0.67$. The trust value reaches a relatively high value in the fourth trust interval.

5.3. Results Analysis

The result calculated from the normalised trust evidence value in the experiment is 0.67. Then, each piece of trust evidence in the security feature S is analysed with the aim of unveiling the nuances of the trust assessment process. First, assuming that the values of P and R are unchanged and that the other values in S are unchanged as well, the value of s_1 increases gradually from 0 to 1 by 0.1 each time. Figure 6 illustrates the change in the trust value. The trust value is 0.67 when s_1 is 0 and gradually decreases to 0.56.

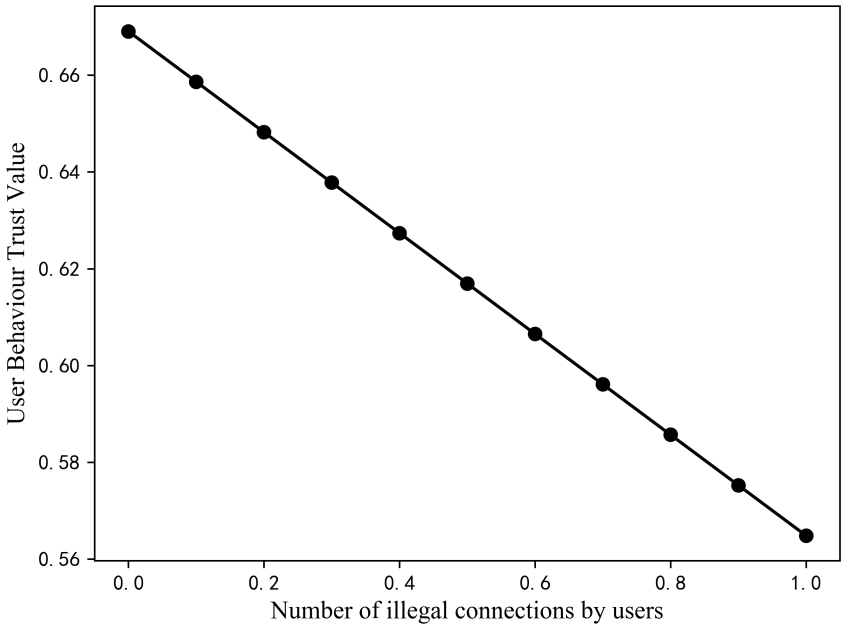

Figure 6. Variation of user trust value with the number of illegal connections.

Assuming that the values of P and R remain unchanged and that the other values in S remain unchanged as well, then gradually increasing the value of s_2 from 0 to 1 with an increase of 0.1 each time, the trust value changes as shown in Figure 7. It can be seen that the trust value is 0.70 at s_2 of 0 and decreases to 0.53.

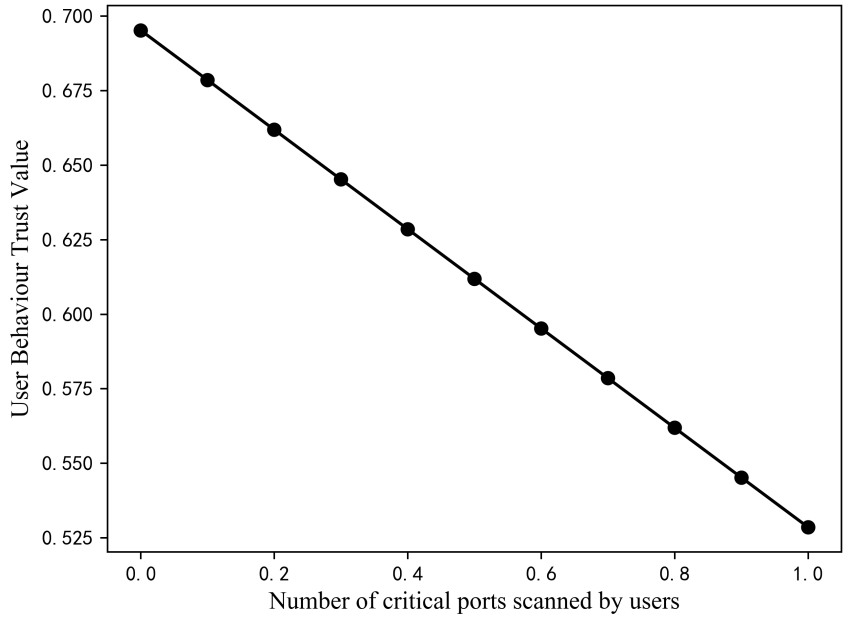

Figure 7. Changes in the number of critical ports scanned by user trust values.

Assuming that the values of P and R remain unchanged and that the other values in S remain unchanged as well, then gradually increasing the value of s_3 from 0 to 1 with an increase of 0.1 each time, the trust value changes as shown in Figure 8. It can be seen that the trust value is 0.67 at s_3 of 0 and decreases to 0.50.

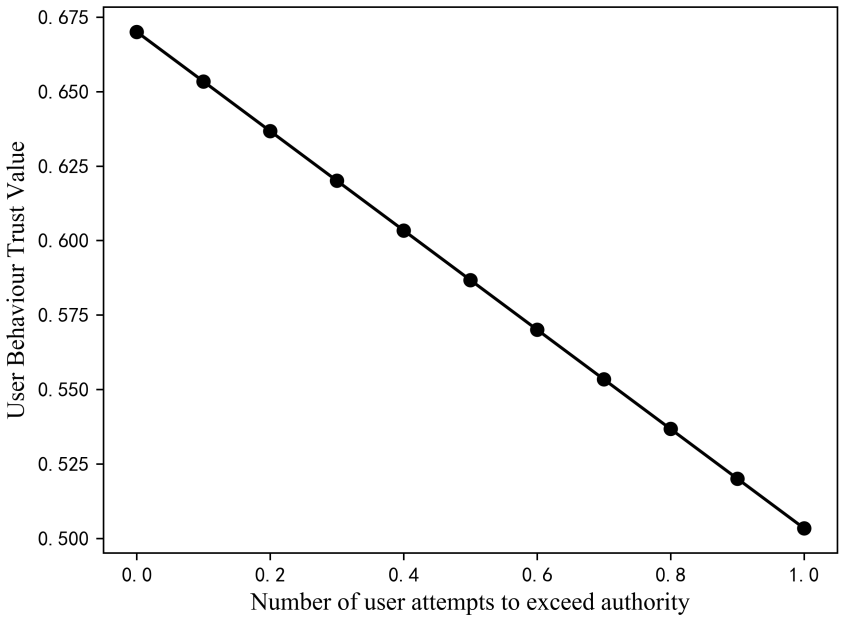

Figure 8. Variation of user trust values with the number of attempted transgressions.

Assuming that the values of P and R remain unchanged and that the other values in S remain unchanged as well, then gradually increasing the value of s_4 from 0 to 1 with an increase of 0.1 each time, the trust value changes as shown in Figure 9. It can be seen that the trust value is 0.69 at s_4 of 0 and decreases to 0.63.

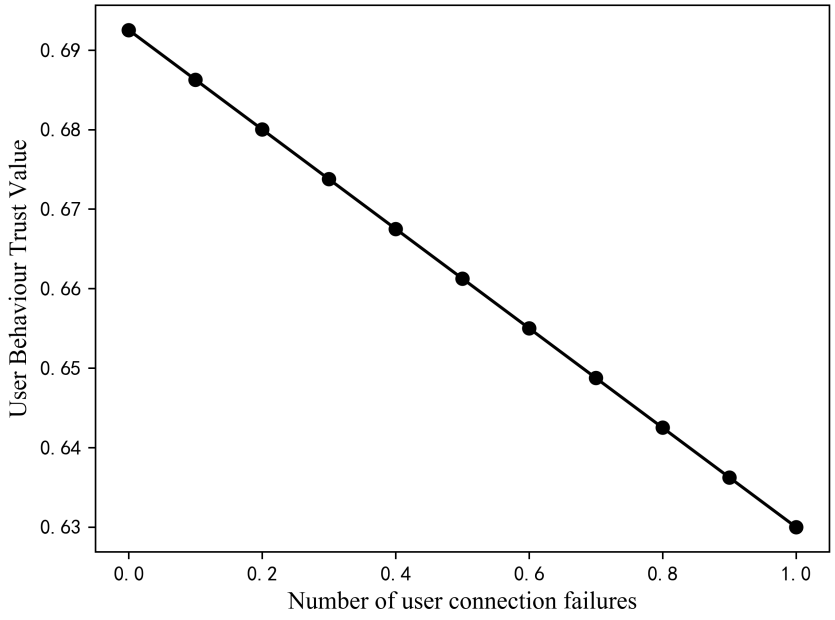

Figure 9. Variation of user trust value with the number of connection failures.

Assuming that the values of P and R are kept constant and the values of all four illegal behaviours in S are gradually increased from 0 to 1, with an increase of 0.1 each time, the trust value changes as shown in Figure 10. The trust value is 0.73 when all values in S are 0, and decreases to 0.23. As the proportion of illegal behaviours increases, the change in the trust value is obvious as it decreases to the untrustworthy interval.

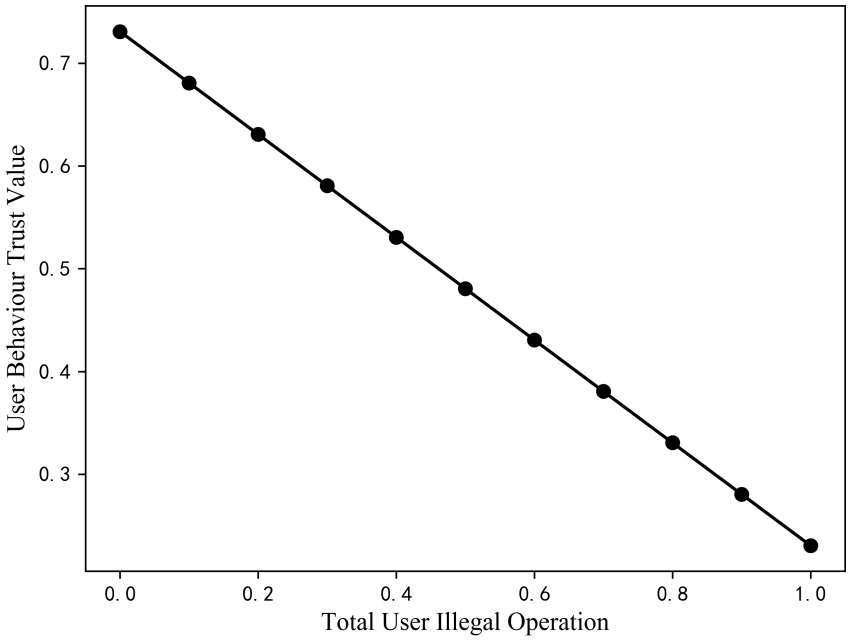

Figure 10. Variation of user trust values with total number of illegal operations.

From the above results, a single change in the security characteristics of the trust evidence does not significantly affect the change in the trust value. Nonetheless, the combination of the four types of trust evidence can significantly change the resulting trust evidence. Meanwhile, the importance of the trust evidence greatly influences the change in the trust value. Among the four values, s_2 and s_3 are more important than the others, therefore, the changes are relatively larger in these cases.

5.4. Validity

In order to prove the validity of the AU-ZTAC model, we analysed it by tracking the changes in the user's trust value at different moments during the user's visit. The changes in the trust value of the user at different moments are shown in Figure 11. The model allows calculation of trust scores based on user behavior and interactions, meaning that a user's trust value changes with time and user behavior. For example, a user may display consistently good behavior over a period of time, accumulating a higher trust score; however, if their behavior suddenly exhibits unusual or suspicious activity, their trust score may drop. This dynamic trust assessment can more accurately capture users' actual intentions and behaviors, thereby enhancing the validity of the model. Only when the user trust value matches the preset threshold for the resource can the user can gain access, thereby better protecting the security of autonomous vehicle networks and user privacy.

5.5. Model Characteristics Analysis

The proposed AU-ZTAC was compared with the access control methods in [9,30,35] in terms of continuous authentication, dynamic access control, fine-grained access control, and access process security. Table 4 shows the comparative results, demonstrating the advantages of AU-ZTAC over other models. AU-ZTAC is based on a zero-trust architecture. Its core feature is its ability to perform continuous verification and authorization, which can effectively solve the problem that traditional perimeter-based network architectures lack effective means of response if attackers break through the boundary and enter the network. In terms of fine-grained access control, AU-ZTAC adopts attribute-based access control. ABAC can accurately control permissions according to actual need, reducing the risk of over-granting permissions. While RBAC privilege assignment is relatively static and role-based, pre-assigning permissions to roles may lead to the network granting too

many or too few permissions. Attribute-based access control is more granular and flexible in privilege assignment than the role-based access control approaches adopted in [9,35]. Regarding access process security, [9] published the access control policy as a smart contract on a blockchain to ensure security and reliability. Through its zero-trust architecture, the proposed model uses a separate control plane for communication in the access control process, which is more secure and reliable compared to the traditional approach presented in [30,35].

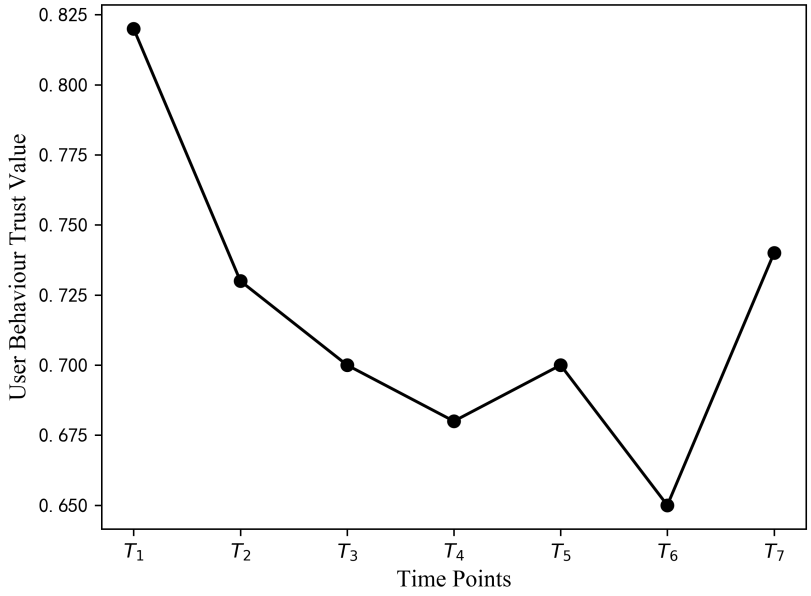

Figure 11. The change in the trust value of the user at different moments.

Table 4. Comparison of AU-ZTAC with other models.

Models	Continuous Authentication	Dynamic Access Control	Fine-Grained	Access Process Security
AU-ZTAC	✓	✓	✓	✓
Reference [9]	×	✓	×	✓
Reference [30]	×	✓	✓	×
Reference [35]	×	×	×	×

6. Conclusions and Discussion

To address the access control challenges in IoV contexts, we propose AU-ZTAC, which integrates a zero-trust architecture with attribute-based access control, leveraging attribute information for precise and nuanced access control. The introduction of trust scores quantifies the trustworthiness of users, necessitating alignment with access control policies and attainment of a trust score meeting the specified threshold before accessing resources. Furthermore, AU-ZTAC recognizes the real-time evolution of user behavior, reflecting changes in trust scores with different user interactions. This adaptability ensures a resilient and responsive security framework, enhancing the overall security and accuracy of access control by comprehensively considering attribute matching and user trust scores. AU-ZTAC establishes a granular access control framework, providing robust defence against both internal and external security threats through a zero-trust paradigm. In this way, it effectively safeguards sensitive data and resources, adeptly addressing contemporary network security challenges.

In our future work, we plan to make improvements in two key aspects: enhancing the trust evaluation mechanism, and integrating anomaly detection and response. First, continuous refinement of the trust evaluation mechanism, possibly by incorporating ma-

chine learning techniques, could enhance the accuracy of user trust scores by adapting to evolving user behaviors and network conditions. Second, integration of anomaly detection and response mechanisms within the zero-trust architecture could fortify the system against emerging threats and contribute to the proactive defense of autonomous vehicles. In a rapidly evolving field, addressing security and privacy issues is critical. Further research in needed improve the safety and reliability of autonomous vehicle networks, leading to safer and more intelligent transportation systems.

Author Contributions: Conceptualization, J.W. and Z.W.; methodology, Z.W.; validation, Z.W.; formal analysis, J.S.; investigation, H.C.; resources, J.W., J.S. and H.C.; data curation, Z.W.; writing—original draft preparation, Z.W.; writing—review and editing, J.W., J.S. and H.C.; supervision, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Major Science and Technology Innovation Project of Shandong Province 2019JZZY010134, Natural Science Foundation of Shandong Province ZR2020MF058, and Shandong Province Science and Technology Innovation Enhancement Project 2022TSGC2544.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. These data are not publicly available due to the experimental data involving another unpublished paper.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study, in the collection, analysis, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

References

1. Alnasser, A.; Sun, H.; Jiang, J. Cyber security challenges and solutions for V2X communications: A survey. *Comput. Netw.* **2019**, *151*, 52–67. [CrossRef]
2. Ghosal, A.; Conti, M. Security issues and challenges in V2X: A survey. *Comput. Netw.* **2020**, *169*, 107093. [CrossRef]
3. Kim, K.; Kim, J.S.; Jeong, S.; Park, J.H.; Kim, H.K. Cybersecurity for autonomous vehicles: Review of attacks and defense. *Comput. Secur.* **2021**, *103*, 102150. [CrossRef]
4. Koopman, P.; Kane, A.; Black, J. Credible autonomy safety argumentation. In Proceedings of the 27th Safety-Critical Systems Symposium, Bristol, UK, 5–7 February 2019.
5. Changalvala, R.; Malik, H. LiDAR data integrity verification for autonomous vehicle. *IEEE Access* **2019**, *7*, 138018–138031. [CrossRef]
6. Badue, C.; Guidolini, R.; Carneiro, R.V.; Azevedo, P.; Cardoso, V.B.; Forechi, A.; Jesus, L.; Berriel, R.; Paixao, T.M.; Mutz, F.; et al. Self-driving cars: A survey. *Expert Syst. Appl.* **2021**, *165*, 113816. [CrossRef]
7. Elkhail, A.A.; Refat, R.U.D.; Habre, R.; Hafeez, A.; Bacha, A.; Malik, H. Vehicle security: A survey of security issues and vulnerabilities, malware attacks and defenses. *IEEE Access* **2021**, *9*, 162401–162437. [CrossRef]
8. Rathore, M.S.; Poongodi, M.; Saurabh, P.; Lilhore, U.K.; Bourouis, S.; Alhakami, W.; Osamor, J.; Hamdi, M. A novel trust-based security and privacy model for internet of vehicles using encryption and steganography. *Comput. Electr. Eng.* **2022**, *102*, 108205. [CrossRef]
9. Wang, H.; Pan, Q.; Guo, K. Access control model based on blockchain and user credit. *J. Comput. Appl.* **2020**, *40*, 1674.
10. Habib, M.A.; Ahmad, M.; Jabbar, S.; Khalid, S.; Chaudhry, J.; Saleem, K.; Rodrigues, J.J.; Khalil, M.S. Security and privacy based access control model for internet of connected vehicles. *Future Gener. Comput. Syst.* **2019**, *97*, 687–696. [CrossRef]
11. Chatterjee, A.; Pitroda, Y.; Parmar, M. Dynamic role-based access control for decentralized applications. In Proceedings of the Blockchain—ICBC 2020: Third International Conference, Held as Part of the Services Conference Federation, SCF 2020, Honolulu, HI, USA, 18–20 September 2020; Proceedings 3; Springer: Berlin/Heidelberg, Germany, 2020; pp. 185–197.
12. Oh, S.R.; Kim, Y.G.; Cho, S. An interoperable access control framework for diverse IoT platforms based on oauth and role. *Sensors* **2019**, *19*, 1884. [CrossRef]
13. Abdul, A.M.; Mohammad, A.A.K.; Venkat Reddy, P.; Nuthakki, P.; Kancharla, R.; Joshi, R.; Kannaiya Raja, N. Enhancing Security of Mobile Cloud Computing by Trust-and Role-Based Access Control. *Sci. Program.* **2022**, *2022*, 9995023. [CrossRef]
14. Belchior, R.; Putz, B.; Pernul, G.; Correia, M.; Vasconcelos, A.; Guerreiro, S. SSIBAC: Self-sovereign identity based access control. In Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 29 December 2020–1 January 2021; pp. 1935–1943.
15. Gupta, M.; Awaysheh, F.M.; Benson, J.; Alazab, M.; Patwa, F.; Sandhu, R. An attribute-based access control for cloud enabled industrial smart vehicles. *IEEE Trans. Ind. Inform.* **2020**, *17*, 4288–4297. [CrossRef]
16. Bhatt, S.; Pham, T.K.; Gupta, M.; Benson, J.; Park, J.; Sandhu, R. Attribute-based access control for AWS internet of things and secure industries of the future. *IEEE Access* **2021**, *9*, 107200–107223. [CrossRef]

17. Challagidad, P.S.; Birje, M.N. Efficient multi-authority access control using attribute-based encryption in cloud storage. *Procedia Comput. Sci.* **2020**, *167*, 840–849. [CrossRef]
18. Ezhil Arasi, V.; Indra Gandhi, K.; Kulothungan, K. Auditable attribute-based data access control using blockchain in cloud storage. *J. Supercomput.* **2022**, *78*, 10772–10798. [CrossRef]
19. García-Teodoro, P.; Camacho, J.; Maciá-Fernández, G.; Gómez-Hernández, J.; López-Marín, V. A novel zero-trust network access control scheme based on the security profile of devices and users. *Comput. Netw.* **2022**, *212*, 109068. [CrossRef]
20. DeCusatis, C.; Liengtiraphan, P.; Sager, A.; Pinelli, M. Implementing zero trust cloud networks with transport access control and first packet authentication. In Proceedings of the 2016 IEEE International Conference on Smart Cloud (SmartCloud), New York, NY, USA, 18–20 November 2016; pp. 5–10.
21. Vanickis, R.; Jacob, P.; Dehghanzadeh, S.; Lee, B. Access control policy enforcement for zero-trust-networking. In Proceedings of the 2018 29th Irish Signals and Systems Conference (ISSC), Belfast, UK, 21–22 June 2018; pp. 1–6.
22. Mandal, S.; Khan, D.A.; Jain, S. Cloud-based zero trust access control policy: An approach to support work-from-home driven by COVID-19 pandemic. *New Gener. Comput.* **2021**, *39*, 599–622. [CrossRef]
23. Guo, B.; Wang, J.; Ma, L.; Zhang, W. Research on Zero Trust Dynamic Access Control Model for Sensitive Data. *Inf. Netw. Secur.* **2022**, *6*, 82–93.
24. Yao, Q.; Wang, Q.; Zhang, X.; Fei, J. Dynamic access control and authorization system based on zero-trust architecture. In Proceedings of the 2020 1st International Conference on Control, Robotics and Intelligent System, Xiamen, China, 27–29 October 2020; pp. 123–127.
25. Lin, L.; Mao, X.; Chu, Z.; Xie, X. Adaptive Access Control for Data Lifecycle in Hybrid Cloud Environments. *J. Softw.* **2023**, 1–20.. [CrossRef]
26. Osborn, B.; McWilliams, J.; Beyer, B.; Saltonstall, M. Beyondcorp: Design to deployment at google. *Security.* **2016**, *41*, 28–34.
27. Ward, R.; Beyer, B. Beyondcorp: A new approach to enterprise security. *Usenix* **2014**, *39*, 6–11.
28. Wang, Q.; Yuan, Q.; Li, F.; Xia, L. Review of zero trust network and its key technologies. *J. Comput. Appl.* **2023**, *43*, 1142.
29. Zhang, J.; Zhang, Z.; Xu, W.; Wu, N. Inter-domain Access Control Model Based on Blockchain. *J. Softw.* **2021**, *32*, 1547–1564.
30. Yu, B.; Tai, X.; Ma, Z. A Study of Attribute and Trust-Based RBAC Model in Cloud Computing Environment. *Comput. Eng. Appl.* **2020**, *56*, 84–92.
31. Hu, V.C.; Ferraiolo, D.; Kuhn, R.; Friedman, A.R.; Lang, A.J.; Cogdell, M.M.; Schnitzer, A.; Sandlin, K.; Miller, R.; Scarfone, K. Guide to attribute based access control (abac) definition and considerations (draft). *NIST Spec. Publ.* **2013**, *800*, 1–54.
32. Shi, J.; Li, R. A Review of Blockchain Access Control in the Internet of Things. *J. Softw.* **2019**, *30*, 1632–1648.
33. Guo, J.; Liu, Z.; Tian, S.; Huang, F.; Li, J.; Li, X.; Igorevich, K.K.; Ma, J. Tfl-dt: A trust evaluation scheme for federated learning in digital twin for mobile networks. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 3548–3560 [CrossRef]
34. Guo, S.; Tian, L.; Shen, X. Research on FAHP in User Behavior Trust Evaluation. *Comput. Eng. Appl.* **2011**, *47*, 59–61.
35. Zhang, K.; Pan, X. Access Control Model Based on User Behavior Trust in Cloud Computing. *J. Comput. Appl.* **2014**, *34*, 1051–1054.