

中文图书分类号: TP391
密 级: 公开
UDC: 004
学 校 代 码: 10005



硕士专业学位论文

PROFESSIONAL MASTER DISSERTATION

论 文 题 目: 基于 UCON 模型的动态风险身份认证
方法研究

论 文 作 者: 刘荣超

专业类别/领域: 计算机技术

指 导 教 师: 沈昌祥 院士

论文提交日期: 2020 年 6 月

UDC: 004
中文图书分类号: TP391

学校代码: 10005
学 号: S201761395
密 级: 公开

北京工业大学硕士专业学位论文 (全日制)

题 目: 基于 UCON 模型的动态风险身份认证方法研究

英文题目: RESEARCH ON RISK BASED DYNAMIC IDENTITY
AUTHENTICATION METHOD BASED ON THE UCON
MODEL

论 文 作 者: 刘荣超

专业类别/领域: 计算机技术

研 究 方 向: 信息安全技术

申 请 学 位: 工程硕士专业学位

指 导 教 师: 沈昌祥 院士

所 在 单 位: 信息学部

答 辩 日 期: 2020 年 5 月

授予学位单位: 北京工业大学

独 创 性 声 明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京工业大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

签 名： 刘荣超
日 期： 2020 年 6 月 5 日

关于论文使用授权的说明

本人完全了解北京工业大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

（保密的论文在解密后应遵守此规定）

签 名： 刘荣超 日 期： 2020 年 6 月 5 日
导师签名： 沈昌祥 日 期： 2020 年 6 月 5 日

摘要

当今,互联网的飞速发展使得互联网用户的数目增长迅速。尽管互联网给人们的生活带来了许多便利,但它也会带来一系列的安全问题。身份认证技术是信息安全领域的重要研究方向,能够在不安全的网络环境中对操作者的身份进行识别,以确认操作者的身份是否合法,从而能够防止非法第三方侵入系统进行危险操作,威胁网络安全。但是,作为维护网络安全的第一道关口,身份认证技术本身也面临着威胁。传统的认证方式已不能很好地适应当前的网络环境,存在着一些缺陷。

传统的身份认证主要存在以下问题:

(1) 灵活性较低。现有方法多依赖用户特有的确定性凭据或特征,通过匹配系统中已有的相关信息,识别该用户是否符合认证的要求。只要用户提供了确定性的凭据,用户的认证结果通常是一成不变的。因此,传统的身份认证在一定程度上缺乏灵活性,无法很好地适应变化的网络环境。

(2) 安全性较弱。在现有的身份认证中,用户的身份信息多由认证中心审核。然而,身份认证中心作为受信任的第三方机构,却不能有效保证它本身的安全性与可信性。例如,当认证中心本身遭到攻击时,用户身份信息可能遭到利用或篡改,甚至带来更大的威胁。

通过变量,UCON 模型能够进行连续的访问控制,满足当前网络环境的基本要求。因此,UCON 模型也被认为是新一代的访问控制模型。但是,UCON 模型也存在一些缺陷。例如,它的安全性亟待加强。为了对网站提供更加灵活和安全的身份认证,本文在 UCON 模型的基础上加以改进,进行了三方面的研究工作。

首先,本文基于 UCON 模型提出了动态风险身份认证方法。当需要对用户进行身份验证时,会同时参考密码验证和权限控制的结果。两种结果共同决定了用户身份认证的结果与对应的访问权限,提高了身份认证的灵活性。

其次,在风险评估阶段,本文提出了基于历史行为的用户风险评估方法。用户的上网行为将被记录,然后进行风险评估,评估得出的风险值和信任值将作为权限控制的依据。

最后,本文利用区块链智能合约,实现用户风险评估及权限控制的自动化运行。同时,利用区块链本身的安全性和不可篡改性,将区块链作为存储相关信息的数据库,从而提高了身份认证的安全性。

论文实现了以上的身份认证及风险评估方法。为验证方案的安全性,使用 SVO 逻辑系统对协议进行逻辑推理分析,并基于各种场景对本方法进行安全性分析。据分析,本方案在安全性上达到了设定的安全目标,因此具有较高的安全

性与可行性。此外，本文以校园网为落地场景，基于提出的方案完成了认证系统的设计与实现，并在模拟校园网络环境上进行了相关实验。实验证明，本方法具有较高的效率和稳定性，可应用于各种需要对用户进行身份认证的场景。

关键词：身份认证；UCON 模型；风险评估；区块链；

Abstract

Nowadays, the rapid development of Internet makes the number of Internet users grow rapidly. Although the Internet brings many conveniences to people's life, it also brings a series of security problems. Identity authentication technology is an important research direction in the field of information security. It can identify the operator's identity in the insecure network environment to confirm whether the operator's identity is legal, so as to prevent the illegal third party from invading the system for dangerous operation and threatening the network security. However, as the first gateway to maintain network security, identity authentication technology itself is also facing threats. The traditional authentication method can not adapt to the current network environment well, and there are some defects.

Traditional identity authentication has the following problems:

(1) Less flexibility. Most of the existing methods rely on the user's unique deterministic credentials or characteristics, and identify whether the user meets the authentication requirements by matching the existing information in the system. As long as the user provides certain credentials, the authentication result of the user is usually unchangeable. Therefore, it can not adapt to the changing network environment, which makes the identity authentication lack of flexibility.

(2) The security is weak. In the existing identity authentication, the user's identity information is mostly audited by the authentication center. As a trusted third-party organization, identity authentication center can not effectively guarantee its own security and credibility. For example, when the authentication center itself is attacked, the user's identity information may be exploited or tampered with, or even pose a greater threat.

Through variables, UCON model can carry out continuous access control to meet the basic requirements of the current network environment. Therefore, UCON model is also considered as a new generation of access control model. However, UCON model also has some defects. For example, its security needs to be strengthened. In order to provide more flexible and secure identity authentication for the website, this paper improves the UCON model and carries out three aspects of research work.

Firstly, a dynamic risk authentication method based on UCON model is proposed. When the user needs to be authenticated, the results of password authentication and permission control will be referred to at the same time. The two results together determine the result of user identity authentication and corresponding access rights, and improve the flexibility of identity authentication.

Secondly, in the stage of risk assessment, a user risk assessment method based on historical behavior is proposed. Users' online behavior will be recorded, and then risk

assessment will be carried out. The risk value and trust value from the assessment will be used as the basis of authority control.

Finally, using the blockchain smart contract, the automatic operation of user risk assessment and authority control is realized. At the same time, using the security and non tamperability of blockchain itself, the blockchain is used as a database to store relevant information.

In order to verify the security of the scheme, SVO logic system is used to analyze the protocol logic reasoning, and the security of the method is analyzed based on various scenarios. According to the analysis, this scheme achieves the set security goal in security, and has high flexibility. In addition, the related experiments are carried out on the simulated campus network environment. Simulation tests show that this method has high efficiency and stability, and can be used in various scenarios where users need to be authenticated.

Keywords: identity authentication; the UCON model; risk assessment; blockchain;

目 录

摘 要.....	I
Abstract.....	III
第 1 章 绪 论.....	1
1.1 研究背景与意义.....	1
1.2 国内外研究现状.....	2
1.2.1 身份认证研究现状.....	2
1.2.2 区块链在身份认证和权限控制中的应用.....	3
1.3 研究内容.....	4
1.4 章节安排.....	5
第 2 章 相关技术综述.....	7
2.1 身份认证技术概述.....	7
2.1.1 身份认证技术概述.....	7
2.1.2 基于行为的身份认证方法.....	7
2.2 UCON 模型概述.....	8
2.2.1 UCON 模型的组成.....	8
2.2.2 UCON 模型的特点.....	9
2.3 区块链技术.....	9
2.3.1 区块链结构.....	9
2.3.2 Hyperledger Fabric.....	11
2.3.3 智能合约.....	12
2.4 安全协议的形式化分析及证明.....	13
2.4.1 SVO 逻辑简介.....	13
2.4.2 符号及语义.....	14
2.4.3 基本定理.....	14
2.5 本章小结.....	15
第 3 章 基于 UCON 模型的动态风险身份认证方法.....	17
3.1 身份认证现状分析及改进思路.....	17
3.1.1 网站身份认证模式现状分析.....	17
3.1.2 改进的动态风险身份认证方法.....	18
3.2 基于 UCON 模型的动态风险身份认证架构.....	20
3.2.1 UCON 模型的引入.....	20
3.2.2 基于 UCON 模型的动态风险身份认证框架设计.....	20
3.3 动态风险身份认证过程.....	21
3.3.1 参数生成.....	21
3.3.2 信息注册.....	21
3.3.3 身份认证.....	22
3.4 动态风险身份认证方案安全性分析.....	23
3.4.1 基于 SVO 逻辑的分析证明.....	23
3.4.2 抵御重放攻击.....	25
3.4.3 动态攻击场景分析.....	26
3.4.4 相互认证.....	27
3.5 本章小结.....	27
第 4 章 基于行为的风险评估及权限控制方法.....	29

4.1 访问行为采集.....	29
4.2 访问行为风险评估.....	30
4.2.1 风险行为等级划分.....	30
4.2.2 访问行为风险评估.....	32
4.3 区块链安全存储与权限控制.....	33
4.3.1 安全存储与匿名保护.....	34
4.3.2 智能合约权限控制.....	34
4.4 区块链应用合理性分析.....	36
4.4.1 区块链安全存储.....	36
4.4.2 智能合约权限控制的优越性.....	38
4.4 本章小结.....	39
第 5 章 系统实现及实验结果分析.....	41
5.1 身份认证系统开发.....	41
5.1.1 环境配置.....	41
5.1.2 系统实现.....	42
5.2 区块链平台相关设计与实现.....	45
5.2.1 认证系统与 Fabric 衔接.....	45
5.2.2 基于 Fabric 的智能合约设计.....	46
5.2.3 智能合约功能性验证.....	47
5.3 实验结果及分析.....	48
5.3.1 实验规模及场景设置.....	48
5.3.2 实验设计及结果分析.....	48
5.4 本章小结.....	51
结 论.....	53
参 考 文 献.....	55
攻读硕士学位期间发表的学术论文.....	59
致 谢.....	61

第1章 绪论

1.1 研究背景与意义

当今,随着互联网的飞速发展,互联网用户数目的增长也愈发迅速。国家自然科学基金会研究表明,全球互联网用户将于2020年达到50亿^[1]。互联网给人们的生活带来了许多便利,但它也给一些不法分子带来了可趁之机。现代网络具有开放性的特点,使得公共信道上存储和传输了许多敏感信息,这样就给非法第三方提供了盗取、篡改甚至攻击的机会。而这些威胁性行为不仅会对用户本身造成影响,也会对提供服务的网站造成破坏^[2]。因此,如何维护信息的安全性一直备受社会关注。

RFC 2828 将用户身份认证定义为验证由系统声明的用户身份的过程^[3]。身份认证技术是信息安全领域的重要研究方向。作为网络安全的第一道防线,身份认证能够在不安全的网络环境中对操作者的身份进行识别,以确认操作者的身份是否合法,从而能够防止非法第三方侵入系统进行危险操作,威胁网络安全。但是,传统的认证方式似乎不能很好地适应当前的网络环境。2018年5月至6月之间,在Akamai Intelligent Edge平台上发生了超过83亿次恶意登录攻击^[4]。据报道,在2018年12月,攻击者能够使用一种技术绕过gmail和yahoo mail等服务提供的双重身份验证^[5]。由此可见,传统的身份认证方式已不能很好地适应当今复杂的网络环境。

虽然传统的身份认证能够在一定程度上验证用户的身份,但仍存在一定缺陷。首先,现有的身份认证方法缺乏连续性与灵活性。现有方法多依赖用户特有的确定性凭据或特征,通过匹配系统中已有的相关信息,识别该用户是否符合认证的要求。另外,当系统对用户进行认证时,是根据用户在某一时刻提供的凭证来判断的,而不能依据该用户之前的行为。其次,现有的身份认证方法由于其中心化的特点,具有较大的安全隐患。在现有的身份认证中,用户的身份信息多由认证中心审核。认证中心作为身份认证的核心机构,其公开性与透明性都是影响身份认证安全性的重要因素。身份认证中心作为受信任的第三方机构,却不能有效保证它本身的安全性与可信性。例如,当认证中心本身遭到攻击时,用户身份信息可能遭到利用或篡改,甚至带来更大的威胁。因此,如果能在去中心化的网络环境中,提供一种灵活且可靠的身份认证方式,将对保护信息安全具有重要意义。

区块链作为一种新型的计算机应用技术,它融合了共识机制、去中心化、加密算法等计算机技术^[6]。如果能将区块链用于身份认证领域,将能够实现新型网络信任机制,创建公开透明、安全可信的网络认证空间。区块链作为一种去中心化的分布式数据库,其链式结构、共识机制等特性,是维护网络数据完整性的重

要保障。区块链重新定义了网络中的信任机制，具有去中心化、安全性、不可篡改性的特点。如果将区块链技术应用于身份认证中，能够实现去中心化的身份认证，使得认证权限分散化，从而降低第三方滥用权力的风险。但是，区块链运行的成本问题也是需要考虑的问题。如果运行成本过高，则对于中小型网络来说，可能会带来较高的成本代价。考虑到此问题，可使用基于智能合约运行的联盟区块链平台，如 Hyperledger Fabric 等，能够在部分可信的环境中运行，具有低成本、低延迟和低带宽密集性，适用于中小型网络。

本文的目的在于，采取更加安全准确的方法验证用户的数字身份，以保护敏感信息，维护网络系统的安全。因此本文基于 UCON 模型，提出了一种基于风险的动态身份认证方法，并在认证系统中引入区块链技术，利用去中心化的网络环境提高身份认证的安全性。本文的研究内容不仅在密码学和信息安全领域具有一定的理论意义，而且对现实生活中新兴的一些电子业务也具有重要的应用价值。

1.2 国内外研究现状

本部分主要介绍国内外对身份认证方法以及区块链在认证领域的研究。首先，简单介绍了身份认证方法的分类，并分别介绍了国内外学者在不同类别的身份认证方法领域的研究。然后，文章介绍了区块链在身份认证和权限控制领域的研究现状。

1.2.1 身份认证研究现状

根据认证的不同因素，身份认证方法可分为以下几种：

- (1) 由用户了解的相关信息进行鉴别 (what you know)；
- (2) 由用户所拥有的相关物品或凭证进行鉴别 (what you have)；
- (3) 由用户的相关生物特征进行鉴别 (who you are) [7]。

密码验证是“what you know”的最常见的例子。与其他方法相比，它的验证方法相对简单，因此，它也通常被认为是现代信息链中最薄弱的环节。人们很少单独使用密码验证，但经常将它与其他方法结合使用。

“What you have”的含义是，人们使用物品来证明他们的身份，如智能卡、U 盾等。将该方法与密码认证相结合，可以提高身份认证的安全性。基于此，Yang 等人^[8]于 2014 年改进了 Song^[9]的智能卡密码认证方法。他们使用随机数代替时间戳，并使用单向散列函数保护密码。但是，该方法计算过程较复杂，因此攻击者可以使用有效的 ID 执行阻塞攻击，从而导致服务器花费大量时间进行无用的计算。Jangirala 等人^[10]提出了一种基于动态身份的智能卡认证方法，用户可以自由选择登录凭证，并随时用智能卡重新生成密码。然而，由于对智能卡的高度依

赖性,如果用户遗失智能卡,用户账户的安全性将面临很大的风险。此外,考虑到更换智能卡的成本,该方法的可扩展性较差。

“Who you are”的身份认证方法比传统的密码认证更加可靠。Odelu 等人^[11]提出了一种多服务器认证的协议,该协议基于生物识别,结合智能卡技术对用户进行认证。在他们的方案中,注册中心(RC)在需要建立密钥时分别对用户和服务器的身份验证。同时,它还支持撤销和重新注册。虽然这种方法能抵抗大部分情况下的阻塞攻击,但该方法仍然面临着智能卡被盗等问题。如果用户注册表受到攻击,系统将面临风险。在 He 的方法^[12]中,基于鲁棒生物特征提出了一种多服务器认证方法,能够在一定程度上保护用户信息免受攻击。但是,一旦生物特征被捕获到系统数据库中,就无法撤销。Sun 等人^[13]提出了一种轻量级的多因素认证方法。他们将用户的生物特征和智能手机的信息结合起来。通过这种方式,可以实现用户和远程服务器之间的相互身份验证。但该方案不保存固定的生物特征模板,它在解决问题的同时增加了计算和通信成本。

总之,“what you have”的身份认证方法取决于用户持有的东西,一旦被盗,就会面临身份被盗等风险。“who you are”取决于认证者的特征,该方法有低成本和过程复杂的问题。此外,这些鉴别方法都是静态的,因此有时可能无法很好地规避网络中的风险。

2013 年,在 RSA 发布的基于风险的认证白皮书^[14]中,描述了一种将用户行为风险分析应用于身份认证方法的基于风险的认证系统。风险分析和评估可用于识别信息资产中的风险,以防止信息受到威胁。Luo 等^[15]设计了一种基于改进 BP 神经网络的信息安全风险评估模型。利用粗糙集来减少影响网络和信息安全的各种因素。通过这种方式,可以实现对风险的评估。但是,该方案没有对风险进行区分,使得评价缺乏全面性。Patil^[16]基于两个安全事件进行了故障树分析。然而,解决方案没有考虑风险的动态性。网络不是一成不变的,风险评估需要反映威胁的变化趋势^[17]。基于风险的认证是一个动态的认证系统,它将访问系统的用户的风险考虑在内。它基于“what you do”,是一种基于行为的身份认证方法,提供了一种新的身份认证思想。然而,该方案的认证过程大多比较复杂,风险评估方法也不充分,需要进一步改进。

1.2.2 区块链在身份认证和权限控制中的应用

区块链技术在本质上是一种去中心化的分布式数据库,同时结合了共识机制和密码学算法等^[18]。进入 2.0 时代后,区块链的高安全性和高扩展性为学者们在身份认证及权限控制领域的研究开辟了新的空间。研究人员提出,可以利用区块链的账务功能,例如,使用数字加密货币等作为身份认证的依据。Sanda^[19]研究

了使用 WiFi 时的身份验证问题。他们选择数字货币作为身份凭证。当用户登录时,只要该用户有虚拟货币,就可以通过身份认证。Raju^[20]的研究基于以太坊匿名账户钱包。它不仅通过公钥地址管理用户身份,还通过智能合约实现网络认证和支付功能。与此同时,一些研究人员致力于构建基于区块链的访问机制。它允许合法用户访问系统中的资源,并且禁止未经授权的用户访问。Cruz JP 等人^[21]将区块链集成到 RBAC 模型中,解决 RBAC 中的跨组织访问问题,实现跨组织认证。Dorri A 等人^[22]将区块链应用于智能家居。他们将访问控制策略存储到区块链中,并通过区块链事务分配权限,以便保护家庭中智能家居的所有通信。Zyskind 等人^[23]描述了一个分散的个人数据管理系统。他们实现了一个可以将区块链转换为自动访问控制管理器的协议。然而,由于现有区块链多是根据算力来解决区块数据一致性的问题,直接利用区块链底层结构进行访问控制付出的成本过高。

智能合约^[24]是一个在区块链上自动运行的脚本。它被描述为“通过计算机执行合同的交易协议”。Azaria 等人^[25]在以太坊的基础上,借助智能合约实现了访问及控制医疗数据,并提出了 medrec 框架,该框架将智能合约与访问控制结合起来,实现了分布式医疗数据与权限的整合。Ramachandran^[26]使用智能合约和 OPMS 来记录不可变的数据路径以及数据的更改,使得收集的数据源可信。

作为一种通过不可变的分布式账本来保证数据完整性的技术,区块链的链式结构,共识机制等特性,是维护网络数据完整性的重要保障。区块链提供的分布式域名服务,有助于缓解当前的 DNS 漏洞,包括 DDoS 攻击、DNS 欺骗等,有利于提高网络系统的安全性^[27]。另外,基于智能合约运行的联盟区块链平台,如 Hyperledger Fabric 等,能够在部分可信的环境中运行,具有低成本、低延迟和低带宽密集性,适用于中小型网络^[28]。基于以上特点,将区块链用于身份认证及权限控制领域,新型网络信任机制通过区块链来实现,有利于网络空间构建新型信任关系。需要注意的是,在私有链和联盟链中,需要确保节点身份的真实性。因为过高的匿名性可能会保护恶意攻击并侵犯用户资源。

1.3 研究内容

针对上述问题和难点,本文在 UCON 模型的基础上,与区块链相结合,提出了一种基于风险的动态身份认证方法。研究内容包含以下几点:

(1)提出了基于风险的身份认证方法。当需要对用户进行身份验证时,参考用户密码认证和访问权限的结果。以上结果共同决定了用户是否可以通过认证,继而进行访问资源。

(2)在风险评估阶段,本文提出了用户行为风险评估方法。根据用户历史行为,可计算出用户的风险值及信任值。用户的风险值和信任值是动态变化的,随用户访问行为的变化而变化。

(3)将区块链应用于用户权限控制环节。用户的风险值和信任值将作为权限控制的依据。将用户的相关信息写入区块链,并通过智能合约控制用户的访问权限。通过这些方法,可以提高认证过程的安全性。

1.4 章节安排

本文共包括5章,其中,第1章描述了研究背景及意义,包括身份认证方法以及关于区块链在权限控制中的应用的相关研究,继而引出研究目的及意义,并对研究内容进行了概要描述。

第2章对涉及的知识及技术进行了叙述。首先,介绍了身份认证技术的概念,描述了基于行为的身份认证方法。然后对UCON模型进行介绍,描述了其基本结构和主要特点;接着介绍区块链技术,介绍了区块链的基本结构和智能合约,尤其是Hyperledger Fabric中链码的工作方式。最后,介绍了协议形式化证明的相关知识,对证明方法和原理进行了论述。

第3章主要描述了基于UCON模型的动态风险身份认证方法。首先,根据现在的身份认证特点进行了需求分析,接着提出了设计思路。然后,提出了方案的整体架构。接下来,对方案的注册过程及认证过程进行了进一步描述与分析。最后,本文对方案的认证过程进行了安全性分析,论证了本方案的安全性与相关优势。

第4章主要叙述了基于行为的风险评估与权限控制方法,首先介绍了用户风险行为的记录方法,即利用Session记录用户风险行为。然后根据风险评估标准,提出了信任值与风险值的概念,并介绍了风险评估的方法。接着,介绍了利用区块链进行安全存储,提出了保护匿名性的方案,并对利用智能合约进行权限控制的方法进行了描述。最后,分析了利用区块链技术的合理性与可行性。

第5章是实验章节,首先,对第3章提出的基于UCON模型的动态风险身份认证系统进行了实验环境的搭建及功能实现。另外,基于Hyperledger Fabric区块链平台,对第4章提出的基于区块链的权限控制方法,进行了智能合约的编写及部署,并测试及验证了链码功能。最后基于校园网的登录场景进行实验,成功测试了系统功能及相关性能,证明了本方案在效率损失在可容忍的范围内,且能够较好地提高身份认证的安全性与灵活性。

第 6 章结论部分对本文提出的基于 UCON 模型的动态风险身份认证方案进行了总结。指出了论文的主要研究内容、成果和创新点，并进一步展望了未来该研究可以开展的工作。

第2章 相关技术综述

本文涉及的相关技术将在本章中进行介绍。首先，介绍身份认证技术，说明身份认证技术的概念，并对基于行为的身份认证方法进行了描述；然后，介绍 UCON 模型，叙述该模型的结构和工作原理。接着，介绍区块链技术，介绍区块链的基本结构和智能合约的工作方式。最后，介绍了协议形式化证明方法和基本原理。

2.1 身份认证技术概述

本节主要阐述身份认证技术的概念及相关知识，通过这些对身份认证技术有一个基本的认知。

2.1.1 身份认证技术概述

身份认证技术^[29]是证明被认证的对象身份是否属实或在有效期内的一种方法。基本思想是通过验证对象的相关属性，以此证明该对象身份的有效性。目前的主要认证依据主要包含以下几种：由用户了解的相关信息鉴别用户身份；由用户拥有的相关物品或凭证鉴别用户身份；由用户的相关生物特征鉴别用户身份。

当前的身份认证方法主要包括：利用密码进行身份认证；使用 PKI 数字证书的鉴别方法；使用持有的物品如智能卡等进行身份认证；使用相关生物特征如指纹等的身份鉴别方法；多因素身份认证等。密码认证最简单，其应用范围也最广泛；数字证书是依靠可信的第三方进行身份认证；生物特征认证根据用户所具有的独一无二的身体特征，通过分析及比对系统中保存的样本以进行认证^[30]。另外，将以上技术中的两种或几种综合起来，便成为了多因素认证。

2.1.2 基于行为的身份认证方法

基于行为的身份认证^[31]基于“what you do”的思想，即将用户行为作为身份鉴别的凭证。它的主要思想来源于用户的行为习惯及规则，若用户的行为违背了其行为特征，则该用户账户可能遭到挟持或顶替。

基于行为的认证过程示意图如图 2-1 所示。

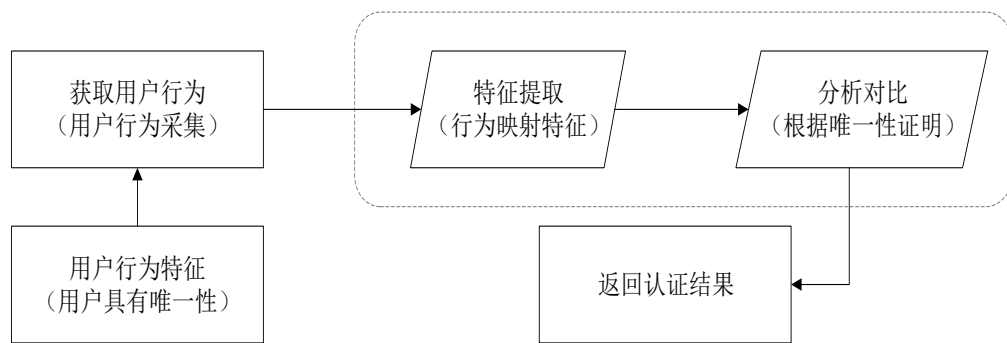


图 2-1 基于行为的身份认证方法

Figure 2-1 Authentication Method Bases Behavior

用户的行为习惯具有独特性和规律性，不能轻易被模仿或替代。基于此，基于行为的身份认证技术不易受到各种攻击的影响，是一种较为安全可靠且准确的身份鉴别方式。

2.2 UCON 模型概述

UCON 模型^[32]是一种新的访问控制模型，它从多个方面扩展了传统的访问控制模型，在本节中，将对该模型相关知识进行介绍。

2.2.1 UCON 模型的组成

UCON 模型由八种元素构成，如图 2-2 所示，包括：主体、主体属性、客体、客体属性、权限、授权、义务和条件。其中，授权基础因素包括主体、客体以及相关属性，在授权的过程中将产生条件、义务及相关授权。义务由主体在访问前或访问过程中履行；基于主体和客体的属性授予访问权限；条件则是系统的环境约束^[33]。

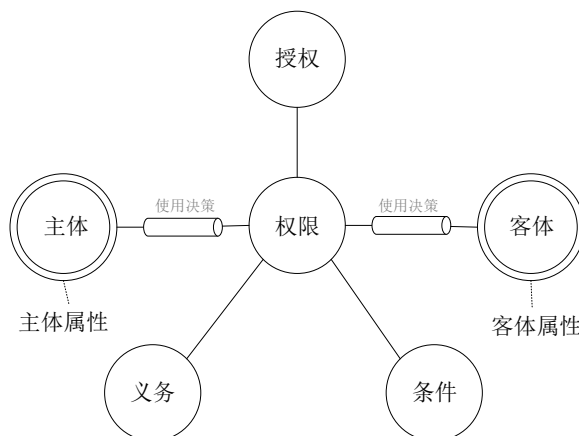


图 2-2 UCON 模型组成元素

Figure 2-2 UCON Model Elements

在传统的访问控制模型中，访问权限通常只授予一次。然而，在 UCON 模型中，授权基于主客体的属性及其所涉及的权利，只有当主体申请访问时，授权才进行。同时，UCON 模型中的授权可以根据对象属性和权限的细分程度，实现更细粒度的权限控制。

责任主要是指，在访问对象资源之前或过程中，根据使用决策主体必须实施的具体操作。例如，当访问者想要访问其个人相关信息时，他们需要提供特定的身份信息。因此，相比传统的访问控制来说，过程责任的设置使得访问控制更具有时效性。

条件主要用来描述系统因素，系统因素不受具体主体的控制。因素的变化（如系统时间、系统负载等）不影响主体和客体的属性。

综合主客体相关属性以及权利、条件等因素进行授权决策，能够使得权限控制更加细粒度化，在控制上也更具连续性。因此，基于 UCON 模型的权限控制能够更加适应有限实名网络环境下个人信息保护的需求。

2.2.2 UCON 模型的特点

UCON 模型具有决策的连续性和属性的可变性的特点。连续性是说，控制决策并不一定要在用户请求访问时才执行，而是可以发生在用户访问过程的任何时刻，包括访问前与访问中。

可变性则体现在，在整个访问过程中，主客体的属性能够发生变化^[34]。属性的变化也可以发生在任何时刻。例如，若出现了不合规定的访问，则相关授权可能会被收回。

2.3 区块链技术

区块链以比特币为典型代表，其本质是一种去中心化的分布式数据库。通过区块链，能够实现在去中心化的系统中进行安全交易，核心技术包括共识机制、链式结构、数据加密以及智能合约等。区块链技术的应用，能够解决现存的中心化机构效率低，安全性差，成本高的问题。

2.3.1 区块链结构

区块链的概念首次出现在人们的视野里，是来自一位化名为“中本聪”的学者的著作^[35]。区块链从狭义上讲，是一种实现了去中心化的存储交易信息的账本。利用区块链能够实现账目信息的安全存储。从广义上讲，区块链是一种分布式防篡改的数据库。区块链利用其链式结构以进行数据存储，使用密码学的相关知识

来保证数据不受到篡改，通过共识机制增加新的区块以更新数据，利用智能合约实现对数据的操作。

区块链架构从上到下分别为提供应用服务的应用层，提供中间协议的智能合约层、激励层、共识层，组成基本网络的网络层与数据层^[36]。其结构框架如图 2-3 所示。

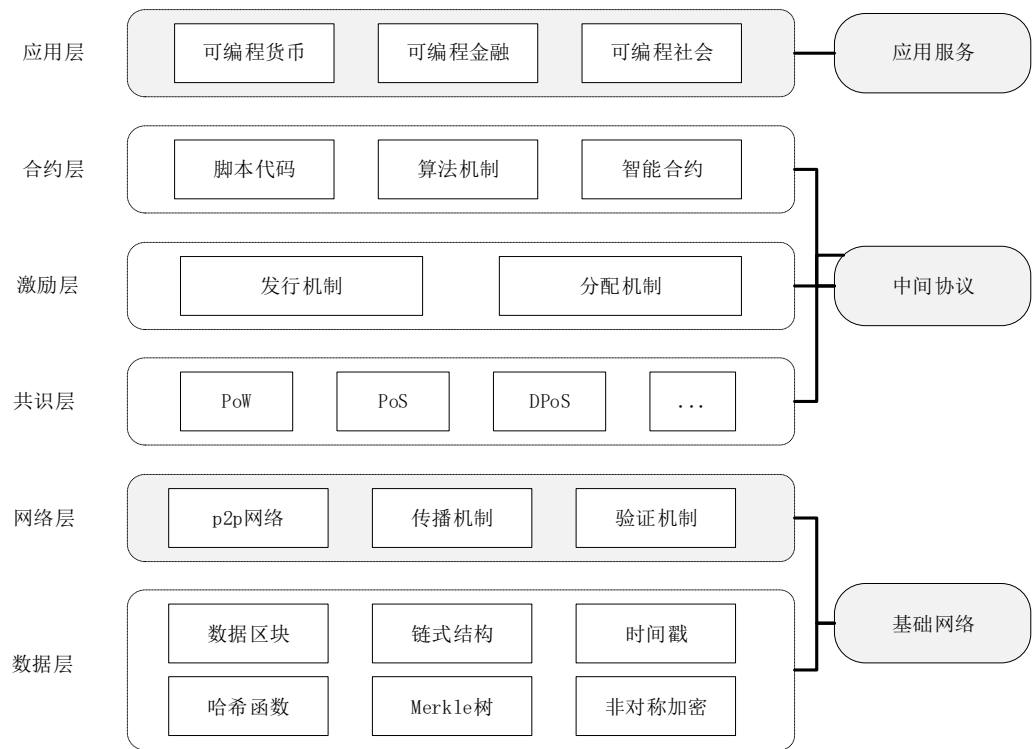


图 2-3 区块链结构图
Figure 2-3 Blockchain Structure

(1) 数据层：区块链的数据层中封装了相关的数据和底层算法。区块链的数据信息和相关技术都在数据层中被封装。为确保封装数据的安全不被篡改，采用共识算法维持数据的一致性。

(2) 网络层：从本质上来说，区块链的网络层是一个 P2P 网络，数据传输能直接在节点之间发生，而无需中心化的第三方服务器。整个区块链系统由所有节点共同维护。

(3) 共识层：区块链的共识机制封装在共识层中。所谓共识，就是区块链网络中的节点间需要依据一种一致性规则，并根据此规则生成新的区块，以进行区块链的维护。

(4) 激励层：激励层封装区块链的激励机制。激励机制就是激励网络中节点工作的机制。通过这种机制，可以促使节点主动参与区块链的更新与维护工作。

(5) 合约层：区块链的可编程化实现与合约层联系密切。该层封装区块链不同功能的智能合约。如果智能合约的执行条件被触发，则该合约能够实现自动化执行。

(6) 应用层：封装了区块链的落地场景及应用案例，如银行、教育、医疗等。区块链如果能够得到实际化应用，许多问题都将迎刃而解。

区块链的特征包括分布式、数据追溯、节点共同维护、可编程以及高安全性^[37]。首先是分布式：区块链对数据的各种操作均依赖于其分布式的结构，通过单纯的数学方法建立节点之间互相信任的关系；其次是数据追溯：区块链采用带有时间戳的链式结构实现对数据的存储，从而使得数据具有更强的可追溯性；接下来是共同维护：区块链采用巧妙的激励机制（例如比特币中的挖矿操作），使得区块链中各个节点都能参与到区块的建立与维护中，新区块的增加也依靠共识机制来完成；然后是可编程：区块链技术提供了灵活的脚本代码，支持用户在链上开发智能合约等应用；最后是高安全性：区块链技术使用非对称加密技术来保证数据的完整性，同时利用共识机制保证数据不被篡改，因此具有很高的安全性和可靠性。

2.3.2 Hyperledger Fabric

Hyperledger 是开源区块链和相关工具的总体项目，于 2015 年 12 月由 Linux 基金会启动^[38]，它描述了存在于基础结构中的节点的关系、链码的部署及执行，以及可配置的服务及共识。虚拟货币在 Fabric 中不存在，这也是它与公有区块链的差别。联盟链结合了公有链和私有链的特点，因此在实际应用中越来越受到重视。

本文使用 Hyperledger Fabric 联盟区块链平台。Fabric 通过可插拔模块来实现各种功能，利用现有成熟的技术来组成区块链的结构。Fabric 的主要结构如图 2-4 所示：

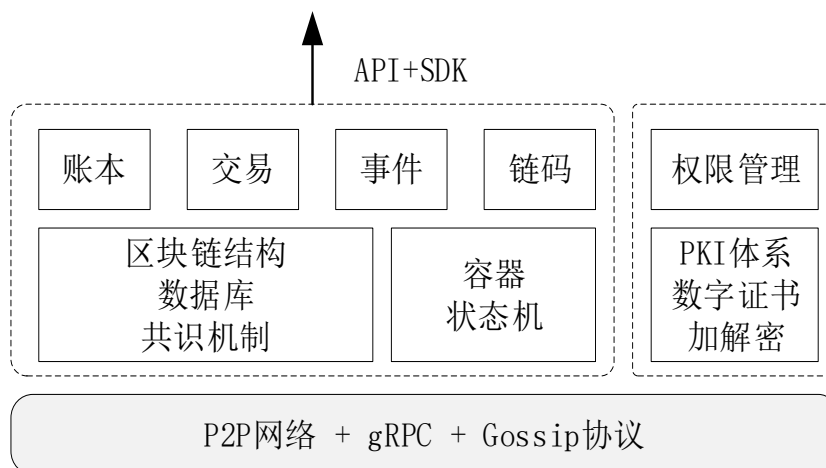


图 2-4 Fabric 基本架构

Figure 2-4 Fabric Basic Architecture

2.3.3 智能合约

智能合约^[39]存储在区块链上，它的定义是一种“能够通过计算机来执行合约相关条款的交易协议”。从本质上讲，智能合约是一段由计算机执行的程序，因此，智能合约也是一种计算机协议。智能合约以代码的形式将合约内容以及执行条件储存在区块链网络中，只要满足执行条件，即可自动准确执行，任何第三方都无法干预。

Fabric 平台中的智能合约称为 Chaincode^[40]，可以用来实现 Fabric 平台上的交易。Chaincode 运行成员一致性的逻辑。Chaincode 通过 Get 和 Put 方法进行 KVS 操作。状态由成员节点（peer）维护，具有永久存储的特点。KVS 中的键名可以识别它们的来源。因此，只有部分 Chaincode 交易能够改变其密钥，从而提高了交易的安全性及可靠性。

智能合约首先按照预设场景进行编写，之后将被部署到区块链上。接下来，智能合约将被打包成 Docker 镜像，在新启动的 Docker 容器中执行实例化，并等待调用。图 2-5 描述了 Chaincode 的安装及部署，以及在区块链中运行的过程。

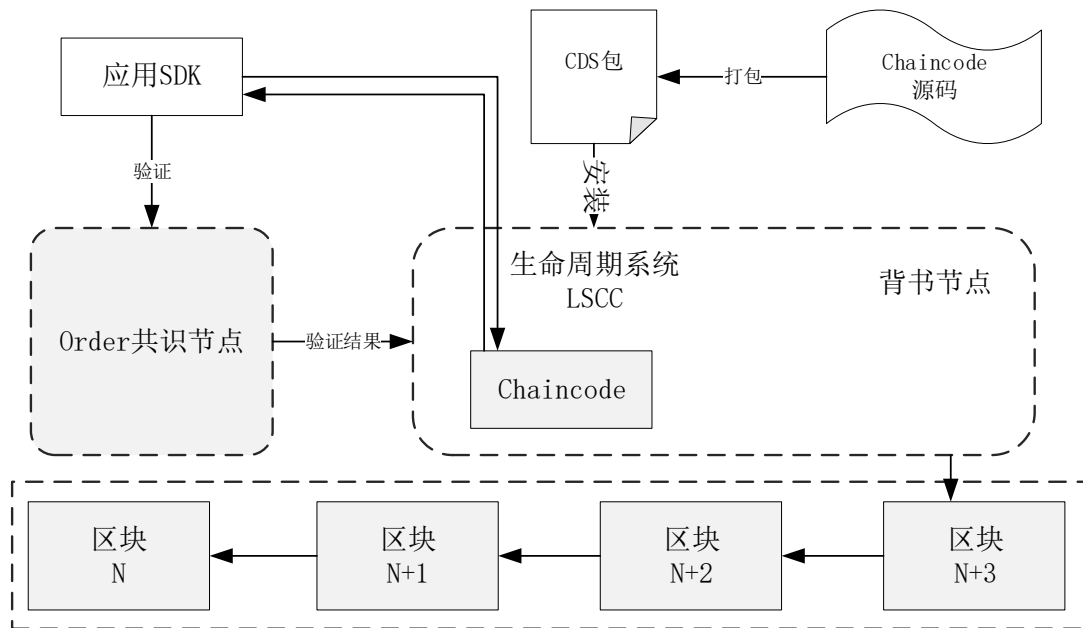


图 2-5 Chaincode 的部署
Figure 2-5 Deployment of Chaincode

Fabric 链码的部署过程主要包括以下几个步骤：

- (1) 由 Fabric 链码的源代码生成 CDS 包；
- (2) 系统链码在区块链节点上安装 CDS 包，在节点上运行的链码也将被生成；
- (3) 通过 SDK 向背书节点发送应用程序请求；
- (4) 链码利用节点进行交易，返回运行结果；
- (5) 运行结果被收集以进行下一步的共识；
- (6) 节点根据共识机制执行共识，生成新块；
- (7) 验证交易，由 fabric 接收交易结果。

2.4 安全协议的形式化分析及证明

模态逻辑技术包括形式化描述及相关推理规则，主要应用于对安全协议的形式化证明^[41]。模态逻辑技术是形式化的分析流程，经过推理及演绎，能够从已存在的命题或信仰推导出新的命题或信仰。模态逻辑技术的推理过程依据相关命题和推理规则。在模态逻辑技术中，BAN 逻辑是最具有代表性且使用最为广泛的逻辑，而其中 SVO 逻辑更为完善和成熟。

2.4.1 SVO 逻辑简介

SVO 逻辑综合了 BAN^[42]、GNY^[43]、AT^[44]、VO^[45]等逻辑系统的特性和优势，是一种成熟可靠而又简单易懂的逻辑推理技术。SVO 逻辑是 BAN 逻辑的拓展。在此处，使用 SVO 逻辑对用户接入阶段进行推理验证，是因为利用 SVO 逻辑可以建立清晰易懂的模型，因此能尽可能避免因为表达式含义不清、逻辑形式混乱或推理规则不明造成的错误。通过 SVO 逻辑对协议进行形式化描述，其逻辑推理和语义解释能够准确理解协议消息的真实含义。

2.4.2 符号及语义

与 BAN 逻辑相仿，在 SVO 逻辑中使用不同的符号描述主体与客体不同的状态^[46]。SVO 逻辑还包括特殊的语义，其相关描述如下所示^[47]：

- (1) M, N : 描述了协议中进行通讯的主体；
- (2) K_m, K_n : 描述了被主体公开的密钥；
- (3) K_m^{-1}, K_n^{-1} : 描述了不被主体公开的密钥；
- (4) K_{mn}, K_{nm} : 描述了通讯过程中主体间的共同密钥；
- (5) A, B : 描述了普通的语句；
- (6) $M \models A$: 表示主体 M 相信消息 A 是真的；
- (7) $\#(A)$: 消息 A 是新发送的；
- (8) $M/\sim A$: 表示在主体 M 发送过的消息中包含消息 A ；
- (9) $M \triangleleft A$: 表示在主体 M 收到的消息中包含消息 A ，且消息 A 能够被主体 M 读取；
- (10) $M \stackrel{K}{\leftrightarrow} N$: 表示主体 M 和主体 N 共同拥有密钥 K ；
- (11) $MK_{(M,K)}$: 表示 K 是主体 M 的公钥；
- (12) $\{A^M\}_K$: 表示利用密钥 K 加密由主体 M 发送的消息 A ；
- (13) $[A]_K$: 表示利用密钥 K 签名消息 A ；
- (14) $SV(A, K, B)$: 表示消息 A 是消息 B 的签名，且可以利用密钥 K 来验证。

2.4.3 基本定理

SVO 逻辑中包含许多基本定理，本小节仅对论文中相关定理进行介绍：

- (1) 信息接收来源定理 A1:

$$(M \stackrel{K}{\leftrightarrow} N \wedge R \triangleleft \{X^N\}_K) \supset (N/\sim X \wedge N \ni K) \quad (2-1)$$

如果主体 R 接收到具有主体 N 签名的消息，并且主体 R 能够用密钥 K 验证签名，则能够证明 R 接收到的是来自 N 的消息。

(2)信息发出定理 A2:

$$| \approx (X_1, \dots, X_n) \supset (M| \sim (X_1, \dots, X_n) \wedge M| \approx X_i) \quad (2-2)$$

在主体 M 发送过的整条消息中, 主体 M 一定发送过这条消息的每一部分。

(3)信息管辖定理 A3:

$$(M| \Rightarrow \varphi \wedge M| \approx \varphi) \supset \varphi \quad (2-3)$$

主体 M 具有其发送过信息的管辖权。

(4)“新鲜的”信息定理 A4:

$$\#(X_i) \supset \#(X_1, \dots, X_n) \quad (2-4)$$

一条新鲜的信息的每一部分必然是新鲜的。换句话说, 若一条信息某一部分是新鲜的, 则这条信息也是新鲜的。

(5)临时值验证定理 A5:

$$(\#(X) \wedge M| \sim X) \supset M| \approx X \quad (2-5)$$

若主体 P 曾经发过消息 X , 且消息 X 是新鲜的, 则主体 P 一定刚刚发送过这条消息。

(6)密钥对称定理 A6:

$$M \stackrel{K}{\leftrightarrow} N \equiv N \stackrel{K}{\leftrightarrow} M \quad (2-6)$$

主体 M, N 之间的通信密钥 K 具有对称性。即如果该密钥是主体 M, N 之间的良好密钥, 则在主体 N, M 之间也具有此种特性。

2.5 本章小结

本章对论文中所涉及到的相关知识和技术进行了叙述。首先, 介绍了身份认证技术的概念及基于行为的身份认证方法; 然后, 对 UCON 模型进行了描述, UCON 模型作为下一代访问控制模型, 对身份认证及访问控制的研究都有着重要意义; 接着介绍区块链技术, 尤其是 Hyperledger Fabric, 智能合约给区块链技术提供了新的应用方向; 最后, 介绍了协议形式化证明方法和基本原理, 使用 SVO 逻辑系统可以对协议的安全性进行推理验证。

第3章 基于 UCON 模型的动态风险身份认证方法

通过对现有身份认证方法特点的研究分析,针对现有身份认证所存在的灵活性及安全性不足等问题,本文基于 UCON 模型,以用户历史行为风险为依据,提出了一种动态的身份认证方法。该方法能够提高身份认证的安全性及灵活性。本章首先根据需求现状进行分析并引出改进思路;其次叙述方案总体架构;接下来对方案的注册及认证过程进行进一步展开分析;最后对方案的认证过程进行了安全性分析。

3.1 身份认证现状分析及改进思路

本节基于现有身份认证方法特征进行分析,并根据现有身份认证方法的缺陷,提出了改进方法并描述了方案的整体结构。

3.1.1 网站身份认证模式现状分析

在网站的运营过程中,存在一些相关的网络资源,为了保护网站资源,需要用户在网站中注册其身份。用户在请求访问资源时,需要通过网站的认证,以证明该用户合法访问的身份。当用户请求登录某网站时,网站对该用户的认证过程如图 3-1 所示。

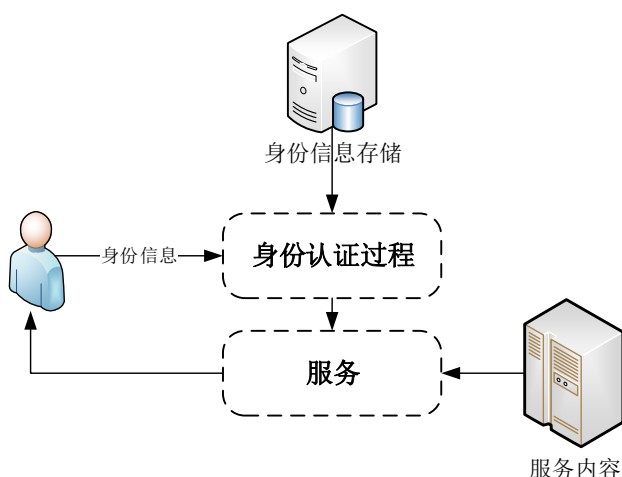


图 3-1 传统方法的认证过程

Figure 3-1 Authentication Process of Traditional Methods

以密码认证为例,用户输入用户名和密码后,系统通过比对用户输入的信息和自己存储的信息是否匹配,便可验证该用户身份的合法性。若用户信息与系统内存储的信息一致,则认为该用户身份合法,身份认证通过。若不匹配,则认为

身份认证失败，用户的访问被拒绝。现有的身份认证方法多依据用户固定的凭据及相关特征，以此来判断用户是否符合认证要求。

由此可见，现有的身份认证方法具有静态性与间隔性的特点。首先，现有方法多依赖比对用户特有的确定性凭据或特征，通过匹配系统中已有的相关信息，即可识别该用户是否符合认证的要求。其次，系统对用户进行认证时，是根据用户在某一时刻提供的凭证来判断的，而不能依据该用户之前的行为。

3.1.2 改进的动态风险身份认证方法

随着移动互联网的迅速发展，越来越多的身份管理需求使得身份认证技术面临着更高的要求，传统的身份认证方法已不能很好地适应变幻莫测的网络环境。以用户访问网站为例，尽管用户仍然能够提供确定的相关凭据，但由于用户的行为是多变且连续的，若其出现过相关风险操作，则依据现有的认证方法并不能很好地识别出风险用户，从而不能可靠地保障通过认证的用户的可信性，进而可能威胁相关网站资源的安全。

从以上存在的问题可看出，只依靠用户具有固定的凭证来鉴别用户的身份是不够的，身份认证还应该结合其他相关凭据。因此，本文在传统的身份认证的基础上，提出了基于用户行为风险授予其访问权限的观点。

对于登录成功的用户来说，每位用户在网站中都有访问行为，通过分析用户的这些行为就可以识别该用户是否出现风险操作。若该用户出现风险操作频率越高越严重，则该用户的风险就越高。对于系统来说，系统对用户的信任程度取决于该用户访问资源时的操作是否符合其正常操作特征。如果当前用户的行为模式与该用户账户以往的行为特征具有比较大的差异，则用户可能不再受到系统的信任。倘若在已有的密码身份认证系统中加入基于用户历史风险行为的二次认证模块，能够较好地提高系统的安全性。

就本文的研究而言，用户的风险不会是一成不变的，出现风险行为的用户受信任的程度更低。另外，用户受信任的程度与时间也有关。用户风险行为离现在越近，风险行为产生的威胁程度越高。最后，用户风险受到环境因素的影响。例如，用户的登录地点及设备都会对其风险产生影响。

基于以上分析，本文的方案使用用户的访问权限动态控制用户的身份认证结果。一个通过身份认证的用户不仅应该通过传统的身份认证，还应该具有正确的访问权限。该权限由用户的风险行为决定。它是动态的，若用户风险过高，该权限将被回收，用户将不能访问网站资源。

综上所述，用户动态风险身份认证方法设计如下：

(1) 用户请求访问网站时, 首先经过密码认证, 其次, 要经过权限认证。只有通过密码认证且具有访问权限的用户才能正常访问网站资源。

(2) 登录成功的用户每次访问网站后, 都会对用户的访问行为进行风险评估, 并得到用户相应的风险程度和受信任程度。

(3) 根据用户风险和受信任程度决定用户访问权限, 作为下次身份认证的依据。

用户访问网站的整个过程如图 3-2 所示。

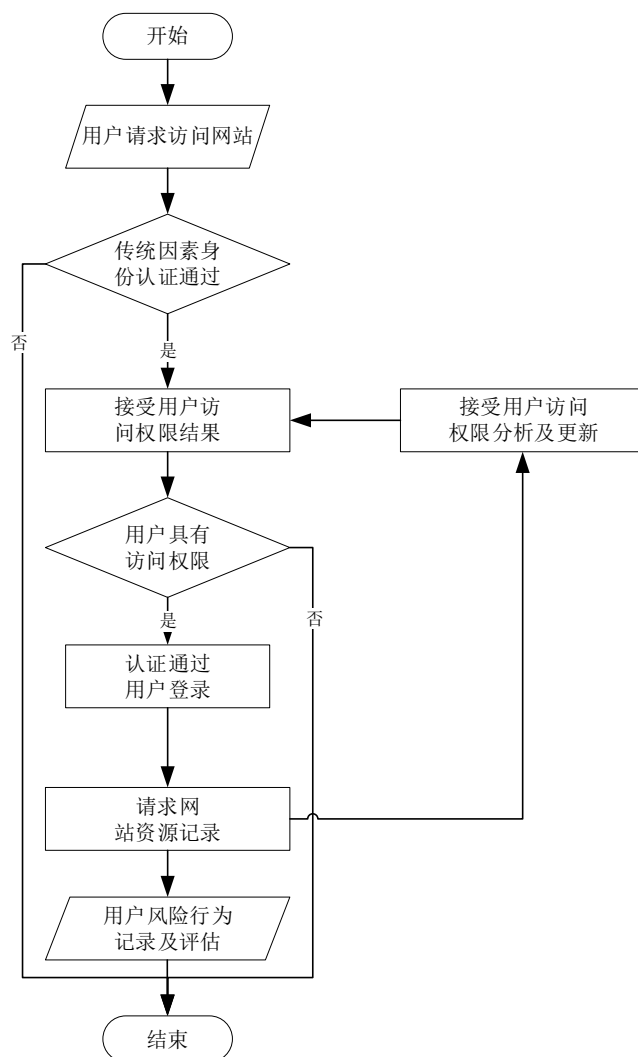


图 3-2 用户访问网站资源过程

Figure 3-2 User Access to Website Resources

相比于传统的身份认证方法, 本文提出的认证方案具有以下优势: 首先, 本方案可以增强现有密码认证方案的连续性, 灵活性和准确性, 且不影响用户的正常使用。其次, 本方案通过结合风险程度与信任程度实现用户权限的灵活可控, 从而更好地适应变幻莫测的网络环境。最后, 本方案将传统的身份认证与访问权限控制相结合, 使得该方案既能满足身份鉴别的需求, 也能应用于实际的落地场景。

3.2 基于 UCON 模型的动态风险身份认证架构

3.2.1 UCON 模型的引入

在身份认证与访问控制模型的选择上，本文引入了 UCON 模型。相比于传统的访问控制模型来说，UCON 模型中的访问权限是不同的。以往的访问控制模型中相关权利是静止的，只要不被撤销，该权限就能由主体一直持有。UCON 模型中则并非如此。相关授权只发生在访问中，而并非是长期持有的。基于 UCON 模型的这一特点，相比于传统的访问控制模型，UCON 模型的动态性使其更符合现代网络动态性与变化性的特点，更加有效地保护了相关信息。

UCON 模型使用动态的授权方式，从不同角度进行决策，能够结合主体与客体的动态性与可变性。在 UCON 模型的基础上，本文设计出用户访问网站资源的访问控制模型。

3.2.2 基于 UCON 模型的动态风险身份认证框架设计

结合第一节所述思路以及 UCON 模型的特点，本文提出了基于 UCON 模型的动态风险身份认证方法。只有通过身份鉴别的用户才能访问网站相关资源。用户的身份认证结果由密码认证结果和访问权限结果共同决定。只有通过身份认证后才能访问相关资源。另外，用户的访问权限不是一成不变的，若用户风险过高，将会失去访问权限。

本文基于 UCON 模型，设计了一种基于 UCON 模型的动态身份认证方法，其整体框架如图 3-3 所示。

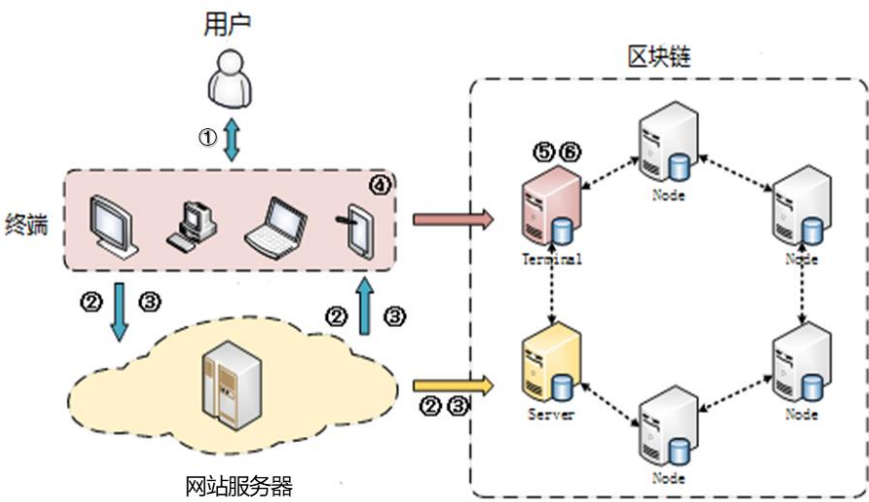


图 3-3 研究方案整体结构

Figure 3-3 Overall Structure of Research Scheme

研究方案的基本框架由客户端、身份认证服务器和区块链三部分组成。如上图所示,整个过程由6个阶段组成:①初始化。本阶段将创建后续过程所需的参数。②注册。在该阶段用户将在系统中注册其身份信息。③认证。服务器在用户请求访问时对其进行身份验证。④用户访问资源。登录成功后,用户可以访问网站资源。⑤风险评估。在此阶段,对用户风险行为进行风险评估。⑥权限控制。在这个阶段,用户的相关信息存储在区块链中,用户的访问权限将由智能合约控制。

3.3 动态风险身份认证过程

本节以用户请求访问网站资源为例,基于用户注册及用户请求访问网站的过程,对本文中基于 UCON 模型的动态风险身份认证过程进行了详尽的描述。首先,在初始化阶段,详细介绍了生成的相关参数,其次,在注册阶段,介绍了本方案中用户注册的详细步骤,以及用户初始访问权限的生成与存储。最后,在身份认证阶段,介绍了本文提出的动态风险身份认证过程,包括密码认证、权限读取及返回最终认证结果。

3.3.1 参数生成

在此步骤中,系统将对后续所需要的参数进行初始化,包括生成用于通信的加密密钥 k 和时间戳 T ,用于保护用户敏感信息的单向散列函数 H 以及用于签名过程的安全参数 λ 。

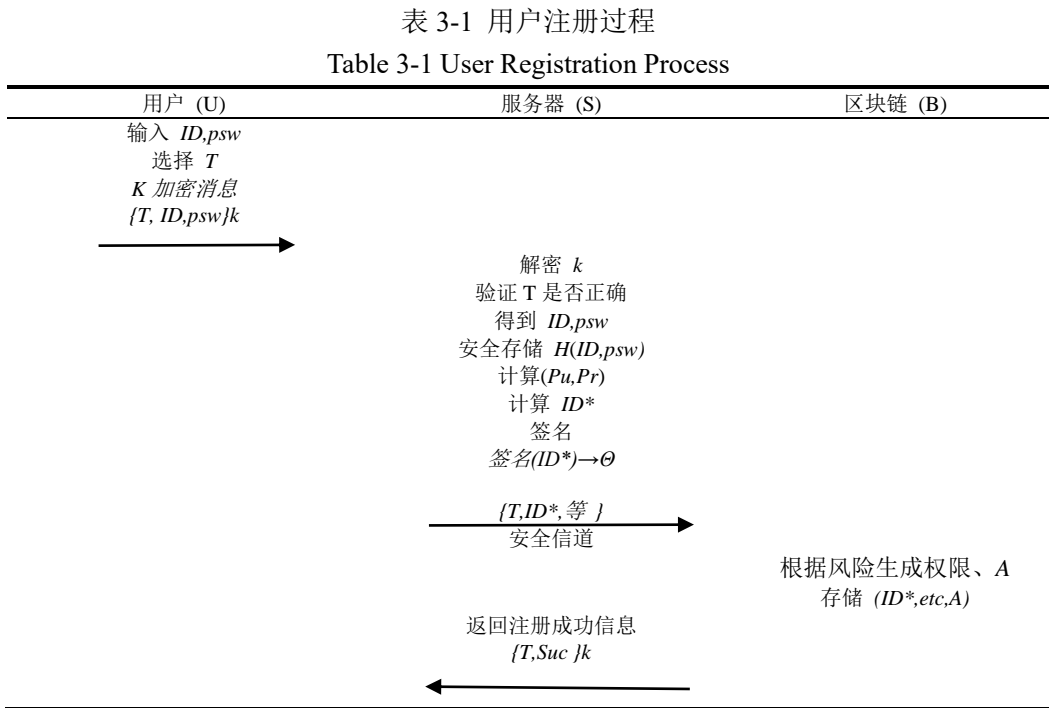
3.3.2 信息注册

用户应在服务器注册其身份信息,以进行后续的访问。在注册时,用户的用户名密码等相关信息将存储在服务器相关存储空间中,与此同时,为了便于后续进行动态的身份认证,本阶段将为用户生成一个初始权限值,并存储在区块链中。此外,为了保护用户的匿名性,区块链上不会存储用户的真实姓名。系统将根据用户名为其生成一个代名,并对其进行签名。用户注册身份的主要过程描述如下:

- (1)用 k 对用户 ID 、密码 psw 和时间戳 T 进行加密,然后发送给服务器 S 。
- (2)服务器在接收到用户的消息后,首先用基于密码的消息认证码(CMAC)^[48]验证密钥 k ,然后计算当前时间与时间戳的差来验证时间戳 T 。如果验证均正确,服务器将提取用户的注册信息。
- (3)服务器使用哈希函数 H 将用户敏感信息转换为固定长度的 Hash 代码,并将其进行安全储存。

- (4) 服务器根据用户 ID 、密码 psw 和安全参数 λ 生成系统签名公钥 Pu 和系统签名私钥 Pr 。
- (5) 服务器将用户 ID 转换为他的代名 ID^* 。
- (6) 服务器使用 Pr 对 ID^* 签名，并获取它的签名 Θ 。
- (7) 服务器通过区块链提供的安全通道发送 ID^* 、用户信息和时间戳 T 。
- (8) 区块链存储用户的初始权限 A 。对于没有进行过访问的新用户，则默认其是可信的，并根据其身份赋予其对应的访问权限。
- (9) 服务器向用户返回成功的注册消息。

用户注册流程如表 3-1 所示：

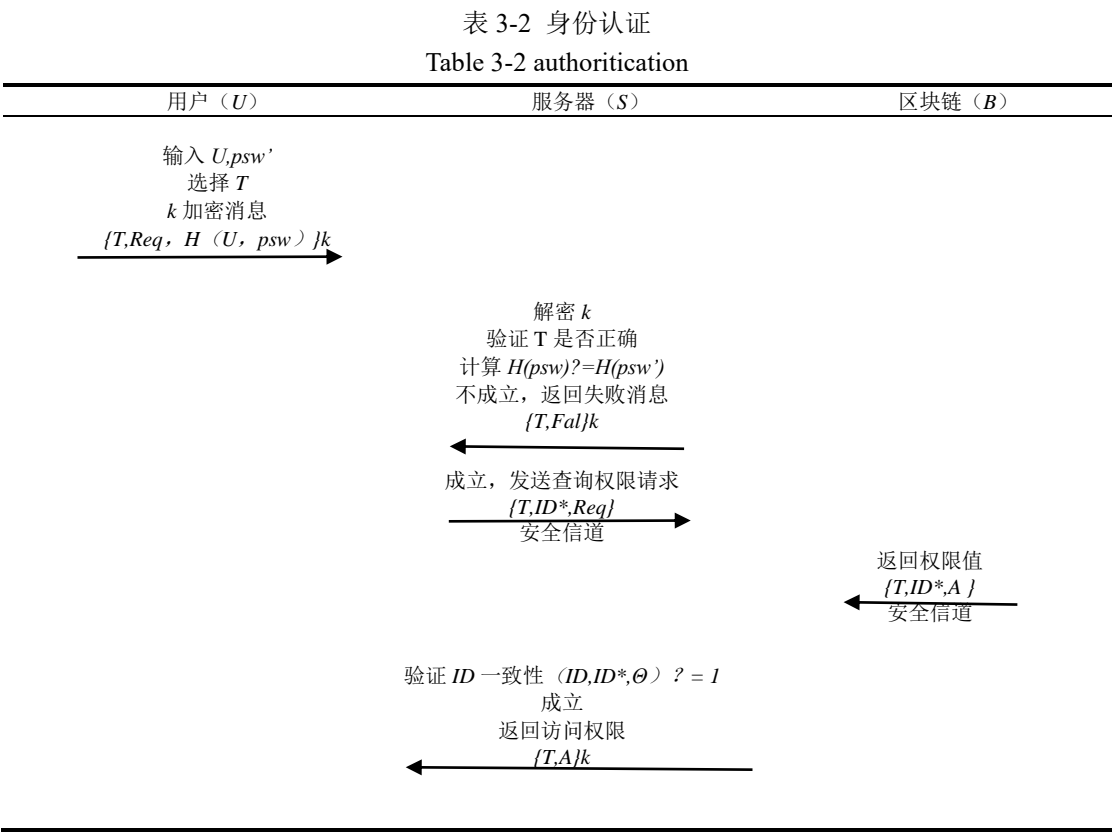


3.3.3 身份认证

这里采用了本文前面提到的动态风险身份认证的方法。用户身份认证的凭据除了用户名和密码外，还要参考根据用户风险得出的权限值。用户的权限值是动态的，它随用户风险的变化而变化。用户身份认证的依据不再单独依靠固有的静态特征，动态变化的权限值在认证中也起到了至关重要的作用。在用户请求访问网站的过程中，服务器对用户的认证过程描述如下：

- (1) 用户输入他的 ID 和密码。由于密码是用户登录时输入的，因此不确定输入值是否与存储值一致。为了与服务器中密码区分，用 psw' 表示用户登录时输入的密码。然后用 k 加密它们和时间戳 T ，并将消息传输到服务器。

- (2) 在接收到来自用户的消息后，服务器首先验证加密密钥 k 和时间戳 T 。如果两者都正确，则可以获得用户登录信息。
- (3) 服务器比较 $H(ID,psw)$ 和 $H(ID,psw')$ 以验证 ID 和 psw' 是否等于存储值。如果验证一致，则表明密码验证成功，然后进行权限验证，否则服务器将返回失败消息 Fal 。
- (4) 服务器向区块链发送查询请求以进行权限验证。当接收到区块链发送的访问权限时，首先执行签名验证算法来验证 ID 与 ID^* 之间的对应关系。如果验证成功，则表示代名 ID^* 属于 ID 。
- (5) 服务器读取从区块链接收到的结果。区块链根据用户风险授予该用户相应的访问权限。用户风险越低，用户具有的访问权限越高。
- 认证流程如表 3-2 所示。



3.4 动态风险身份认证方案安全性分析

3.4.1 基于 SVO 逻辑的分析证明

本节使用 SVO 逻辑进行推理，以证明协议的安全性。通过 SVO 逻辑能够形式化协议。将形式化的协议加以推理演绎，不仅能够证明协议是否正确，还能够帮助使用者发现协议的缺陷与不足。

对第二阶段的认证过程进行模型化,用 U 代表用户, S 代表服务器, Tu , Ts 代表时间戳, K 代表 U 与 S 的共享通信密钥。用户输入的信息用 (IDu, PWu) 描述,并用散列函数 H 进行加密,对于密码校验与权限查询得到的最终结果,把它记为 Ns 。则得到如下流程,消息 1 如公式 (3-1) 所示,消息 2 如公式 (3-2) 所示:

$$U \rightarrow S: \{H(IDu, PWu), Req, \#(Tu)\}k \quad (3-1)$$

$$S \rightarrow U: \{(Ns), \#(Ts)\}k \quad (3-2)$$

认证阶段的目标是确信用户将收到可信的认证结果。而必要的条件是一方相信另一方能够控制自己生成的消息,他们都相信每个人都有一个正确的密码密钥来保证整个通信的安全。基于此,将目标和初步假设描述如下:

目标如公式 (3-3) 所示:

$$U| \equiv Ns \quad (3-3)$$

初始假设如公式 (3-4) 至公式 (3-9):

$$U| \equiv U \stackrel{k}{\leftrightarrow} S \quad (3-4)$$

$$S| \equiv S \stackrel{k}{\leftrightarrow} U \quad (3-5)$$

$$S| \equiv \#(Tu) \quad (3-6)$$

$$U| \equiv \#(Ts) \quad (3-7)$$

$$U| \equiv S \Rightarrow Ns \quad (3-8)$$

$$S| \equiv U \Rightarrow H(IDu, PWu) \quad (3-9)$$

依据初始化假设,使用 SVO 逻辑系统进行逻辑推理,验证用户接入过程的安全性。

由初始假设公式 (3-5), 以及公理 A6 得 R1:

$$R1: S| \equiv U \stackrel{k}{\leftrightarrow} S, S| \equiv S \stackrel{k}{\leftrightarrow} U \quad (3-10)$$

由消息 1, R1, 和公理 A1 得 R2:

$$R2: S| \equiv U| \sim ((IDu, PWu), \#(Tu)) \quad (3-11)$$

由初始假设公式 (3-6), 公理 A4 得 R3:

$$R3: S| \equiv \#((IDu, PWu), Tu) \quad (3-12)$$

由 R2, R3 和公理 A5 得 R4:

$$R4: S| \equiv U| \approx ((IDu, PWu), Tu) \quad (3-13)$$

由初始假设公式 (3-9), R4 和公理 A2, 公理 A3 得 R5:

$$R5: S| \equiv (IDu, PWu) \quad (3-14)$$

则可知 S 正确接收到了用户发来的消息。

由初始假设公式 (3-4), 以及公理 A6 得 R6:

$$R6: U| \equiv U \stackrel{k}{\leftrightarrow} S, U| \equiv S \stackrel{k}{\leftrightarrow} U \quad (3-15)$$

由消息 2, R6, 和公理 A1 得 R7:

$$R7:U| \equiv S|\sim(Ns, \#(Ts)) \quad (3-16)$$

由初始假设公式 (3-7), 公理 A4 得 R8:

$$R8:S| \equiv \#(Ns, Ts) \quad (3-17)$$

由 R7, R8 和公理 A5 得 R9:

$$R9:U| \equiv S| \approx (Ns, Ts) \quad (3-18)$$

由初始假设公式 (3-8), R9 和公理 A2, 公理 A3 得 R10:

$$R10:U| \equiv Ns \quad (3-19)$$

至此, 初始目标得证。从上述 SVO 逻辑分析可以得出, 本方案实现了用户和服务器间的安全认证, 因此具有较高的安全性。

在本小节中, 利用 SVO 逻辑从逻辑角度证明了本方案的安全性。在接下来的小节中, 本文以真实用户访问网站为落地场景, 分析了在不同的场景中的攻击。针对这些攻击者对网站登录认证发起的攻击, 本文提出的认证方案具有相应的应对措施与安全保障, 从而证明了本文提出的方案的安全性与准确性。

3.4.2 抵御重放攻击

重放攻击^[49]是登录认证中最常见的攻击之一, 主要发生在请求访问网络的过程。它是指攻击者截获用户发送的包, 修改后将其重新发送, 以此达到欺骗系统的目的。因此, 本小节中以重放攻击作为攻击方法, 选择用户访问网站时的注册以及认证作为攻击的场景。

场景一: 在接入阶段, 用户 U 向服务器 S 发送接入请求消息 $\{T, ID, psw\}k$ 。攻击者 A 截获该消息并重新发送消息 $\{T^*, ID, psw\}k$, 其中 T^* 是收到此消息的时间戳, 如图 3-4 所示。

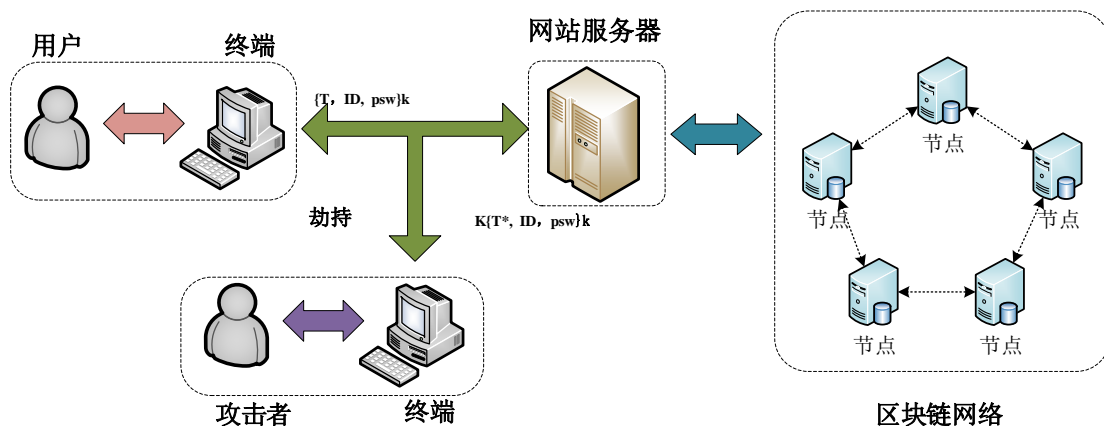


图 3-4 重放攻击场景 1

Figure 3-4 Replay Attack Scenario 1

场景二：如图 3-5 所示，在注册阶段，用户 U 向服务器 S 发送注册请求消息 $\{T, ID, psw\}^k$ 。假设 U 被攻击者挟持， U 将重新发送消息 $\{T^*, ID, psw\}^k$ ，其中 T^* 是第二次发送消息的时间戳。

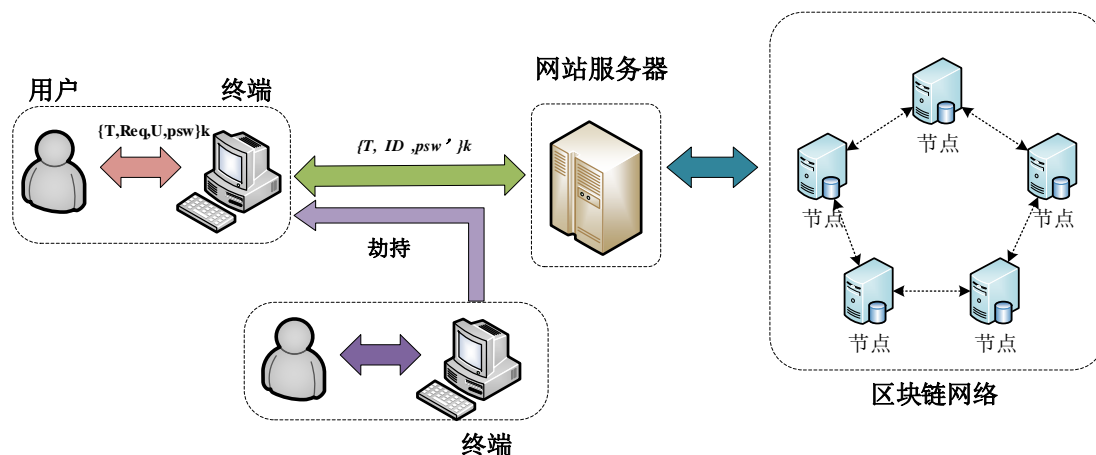


图 3-5 重放攻击场景 2

Figure 3-5 Replay attack scenario 2

安全分析：令 ΔT 为最大传输延迟。如果 $|T - T^*| > \Delta T$ ，则 S 将丢弃该消息。同理在用户接收服务器认证结果的消息时，使用时间戳，如果此消息由攻击者重放到 U ，则时间戳验证失败，消息将被用户丢弃。因此，所提出的方案能够抵御重放攻击。

3.4.3 动态攻击场景分析

在本文提出的方案中，用户的权限不再一成不变，他的权限随风险的变化而变化。而用户的权限值直接影响用户身份认证的结果。用户的身份认证从静态的认证方式变为动态的连续的认证方式。基于本方案的以上特点，本节中以用户访问网站时的相关操作作为落地场景，选择方案中出现频率最高，也最为重要的身份认证阶段，在该阶段中，考虑三种攻击场景。

场景一：攻击者 A 使用猜测密码的方式，试图盗取用户 U 的账户。则攻击者 A 的每一次失败的尝试都会被作为该账户的风险行为记录下来。则一段时间后，即使该攻击者 A 最终猜测出用户名密码，该账户也可能由于风险过高失去访问权限。

场景二：攻击者 A 盗取用户密码进行登录，由于攻击者 IP 地址的不同，以及攻击者主机 MAC 地址与原用户是不同的，则这些差异将会被记录，并增加该用户账户的风险。

场景三：攻击者 A 挟持用户 U 的账户进行高危操作，如修改登录密码、用户敏感信息或进行未授权的操作等。该用户账户的这些风险行为都将被记录，若风险过高，该账户将不能进行正常登录。

由此可见，针对于不同的攻击场景，相比于传统的密码认证方式，本文提出的方案能够更好地针对不同情况下的攻击行为做出相应举措。本方案将风险控制与身份认证结合起来，具有更高的安全性。即使用户通过了密码认证，也不能说明用户是可靠的。用户必须具有正常的访问权限，否则仍不能访问资源。

3.4.4 相互认证

本节中第一部分的目标可以证明，在本方案中，用户 U 发送认证请求后，最终能够从服务器 S 获取可信的认证结果。因此，在整个认证过程中，用户 U 和服务器 S 是可以互相信任的。

3.5 本章小结

本章介绍了关于本文提出的基于 UCON 模型的动态风险身份认证方法的相关内容。首先，分析了现有的身份认证方法的特点：现有方法多依据用户固定的凭据及身体特征，以此来判断用户是否符合认证要求。因此，现有的身份认证方法多为静态的。但是，由于用户的操作是复杂多变的，即使该用户通过认证，也不能保证他是绝对受信任的。因此，如何根据用户的变化提供可靠的认证结果是身份认证中的一大问题。针对以上问题，明确了本方案的需求，并提出了本方案的设计思路；其次，基于 UCON 模型，分析了引入 UCON 模型的优势，并对基于区块链的存储模式进行了分层结构设计，分别介绍了每层结构的具体功能；最后基于该模型，结合本文的具体应用场景，设计了基于该模型的动态风险身份认证方法的总体框架，在明确了总体框架之后，详细介绍了网站初始化阶段，阐述了后续所需的参数。其次，描述了用户注册及身份认证过程，具体阐述了基于风险的用户注册及身份认证的具体方法及流程。最后，对该认证方法进行了安全性分析。首先基于 SVO 逻辑，对用户认证过程进行了形式化证明。接着根据常用的攻击场景，对方案进行了分析。可以看出，本文的认证方法相比于传统的认证方法具有较高的灵活性。

第4章 基于行为的风险评估及权限控制方法

本章主要描述了基于行为的风险评估及权限控制方法。用户通过认证后，可以访问网站相关资源。此时用户的访问行为将被记录，并进行风险评估。区块链根据用户风险控制用户访问权限，并安全存储用户相关信息。

4.1 访问行为采集

在经过身份认证后，一个具有正常访问权限的用户可以访问网站上的资源。结合网站的相关特点，可使用 Session 在用户访问期间存储信息。当从一个页面跳转到另一个页面时，访问记录将存储在 Session 中。

在网络应用中，Session 机制^[50]被称为“会话控制”，用来存储用户访问会话的信息。与 Cookie 不同，它是一种将信息存储在服务器端的机制。当用户请求访问应用页面时，系统首先会查询是否为用户创建了 Session 对象，如果没有，将为其创建一个新的 Session。Session 在创建后，若用户在页面之间跳转，Session 中的变量会一直存在于会话中。只有会话过期，Session 中的信息才会被删除。其工作原理如图 4-1 所示。



图 4-1 Session 工作方法

Figure 4-1 Session Working Method

本文的方案利用 Session 的这一特性，标记用户单次的访问行为。借助 Session，可以为该用户创建一个 Session 对象，以标记用户的单次会话，并在整个访问过程（登录→访问→注销）中记录相关信息。其主要执行步骤如下：

（1）若用户成功登录，则服务器会为用户本次的访问创建相关的 Session，且该 Session 由其独特的 Session ID 标记。

（2）利用生成的 Session ID 标识本次整个访问过程。服务器在接收到用户客户端在网页间的跳转请求时，将验证 Session ID 是否存在于请求中。

(3) 若 Session ID 存在于该请求中, 服务器将依据该 Session ID 检索出对应的 Session, 并将相关行为记录到后台。

(4) 如果不包含 Session ID, 则说明该用户已退出登录或访问超时, 此时将拒绝访问, 并跳转回登录界面。

(5) 当用户登出时, 相关风险记录将以 Session ID 为标识, 并交由区块链进行风险评估。

在本文的方案中, 访问过程中用户行为的采集不需要用户主动的配合, 具有非入侵性, 因此用户行为的采集可以在用户操作计算机期间不间断的进行, 为后续用户风险评估提供有力依据。基于此, 本文提出的身份认证方法相较于传统的身份认证方法具有更好的连续性。

4.2 访问行为风险评估

用户在整个访问网站的过程中, 其操作不是一成不变的。本节将对本文提出的方案中的风险评估方法进行详细介绍。若访问网站的用户出现风险操作, 则可能会影响其访问权限, 并进一步影响认证结果。对于用户风险的衡量, 本文根据用户现有操作对其未来的风险进行预测。在本文中, 使用风险值及信任值来描述用户受信任的程度。

4.2.1 风险行为等级划分

用户的上网操作是复杂多样的, 若出现风险操作, 对网站的威胁程度也不尽相同。在本文中, 根据用户的历史行为, 来判断用户的定性风险。定性风险由资产价值、脆弱性和威胁程度构成。根据用户风险行为的构成要素、持续时间和频率, 能够计算出用户的风险值和信任值。

本文以校园网的用户接入过程为场景, 对用户行为风险的计算方法进行了描述。根据《信息安全风险评估规范》(GB/T 20984-2007) 中对风险因子的定义, 可以将校园资源价值、脆弱程度和用户风险行为划分为不同等级。等级划分如表 4-1 至表 4-3 所示:

表 4-1 描述了不同价值水平的校园资源。网站中的资源有许多种类型, 其对于网站的重要程度也不同。例如, 如果网站的操作系统遭到破坏, 重建该网站付出的代价将是比较大的。本文使用校园网作为落地场景, 根据网站资源的不同重要程度, 将校园资源的价值划分为以下五个等级。

表 4-1 校园资源价值水平
Table 4-1 Resource Value Level of the Campus

重要性	价值	类型	介绍
V	80-100	基础设施	由计算节点组成的服务器资源池，如果被破坏将造成非常严重的破坏
IV	60-80	系统资源	网站操作系统，如果被破坏会带来较严重的破坏
III	40-60	共享数据	需要共享的数据，如果被破坏将会造成一些损失
II	20-40	应用工具	网站的辅助应用，被破坏后会造成轻微损失
I	0-20	门户信息	公告，如果被破坏几乎不会造成任何损失

表 4-2 描述了当资源遭到破坏后，所造成后果的严重程度。例如，当风险操作出现时，若只导致了一般的 URL 跳转，则该操作对网站的危害程度是可以忽略不计的。依据不同风险操作对网站的不同危害程度，将造成的后果划分为五个不同等级。

表 4-2 危害程度
Table 4-2 Level of Vulnerability

水平	价值	范围	介绍
V	80-100	非常危险	直接破坏系统权限的漏洞
IV	60-80	危险	未经授权泄露或访问敏感信息
III	40-60	中度危险	导致 Web 应用程序拒绝服务
II	20-40	略微危险	在本地造成拒绝服务或泄漏常规信息
I	0-20	几乎安全	一般的 CSRF 或 URL 跳转

表 4-3 描述了攻击行为的风险水平。例如，在访问过程中出现的一些异常操作、或者是对内容的异常访问等，都将对系统造成一定危害。根据不同的风险行为类型，将风险行为水平划分为四个等级。

表 4-3 风险行为水平
Table 4-3 Level of Risk Behavior

水平	价值	行为类型	介绍
IV	75-100	恶意安全行为	通过病毒、木马攻击系统，可能造成非常严重的破坏
III	50-75	违约行为	未经授权的操作，可能会影响系统的正常运行
II	25-50	异常行为	包括异常操作、异常访问内容等，对系统有一定危害
I	0-25	异常状态	包括异常登录位置、异常设备等，对系统几乎没有影响

根据以上等级划分，可具体计算出用户的风险值。风险值计算方法将在下一节详细描述。

4.2.2 访问行为风险评估

本节描述了用户风险值及信任值的计算方法。用户风险值用于描述用户的行为风险，可以直观地描述用户在每次访问网站后的风险变化。另外，用户的信任值用于长期反映该用户受信任的情况，从而能够长期地反映该用户的风险状况。结合用户风险值和信任值，能够较为准确地反映该用户的行为特征，继而为用户身份认证提供重要依据。

用户风险值的计算方法如下所示：

根据信息安全风险方程，用户风险值 F 可以用公式(4-1)表示：

$$F = W \times L \times R \quad (4-1)$$

W 是资源的价值，如表 3 所示，描述了不同的资源种类和资源的重要性； L 是脆弱程度，如表 4 所示， R 是风险行为对系统的危害程度，如表 5 所示。但是，计算结果的数值过大将不利于描述风险变化的趋势。为了方便地评估风险，可以用公式（4-2）的形式对其进行量化：

$$F = \sqrt[3]{W \times L \times R} \quad (4-2)$$

根据上述公式，可以得到用户在某一时刻的风险，即用户的静态风险。然而，本文认为，用户的行为是不固定的，因此他们的风险也应该是动态的。因此，结合用户风险行为发生的频率和持续时间，可以得到用户的动态风险值，如公式（4-3）所示：

$$F = \begin{cases} \theta \times F_0 & (a) \\ F_0 + e^t \times \sqrt[3]{W \times L \times R} \times \frac{M}{Ti} & (b) \end{cases} \quad (4-3)$$

其中, F 是用户的风险。

在公式(4-3)中, 式(a)反映了用户行为正常时, 用户风险值的衰减过程。 θ 是风险衰减因子, 是一个小于 1 的常数。 F_0 是用户的最近一次的风险值。可见, 当用户行为正常时, 其风险会逐渐而缓慢地减弱。

公式(4-3)中(b)反映了用户出现风险行为时风险值的增加趋势。在这个公式中, t 表示用户风险行为的持续时间, $\frac{M}{T_i}$ 表示这些行为的频率。可以看出, 风险值随着用户风险行为的频度和持续时间的增加而急剧增加。

为了提高风险评估的准确性, 本文还根据用户的信任值来衡量用户的访问权限。可以通过用户的风险值来计算用户的信任值 Tr , 如式(4-4)所示:

$$Tr = \begin{cases} Tr_0 - P^{(F-F^*)}, & (F > F^*) \text{ (c)} \\ Tr_0 + \frac{(F^*-F)}{q}, & (F < F^*) \text{ (d)} \end{cases} \quad (4-4)$$

Tr 表示用户的信任值, Tr_0 是用户最近一次的信任值。 F 是用户的风险值, F^* 是用户的风险阈值。

在公式(4-4)中, 式(c)表示当用户的风险值超过风险阈值时信任值衰减的过程。 P 是控制用户信任度降低程度 1 的修正因子, 它是一个大于 1 的常数。式(d)表示当用户的风险低于风险阈值时用户的信任值增加的过程。 q 是一个大于 1 的常数, 它用来调整信任度的增加程度。

由此可见, 用户信任值与风险值具有以下关系:

- (1) 用户信任值的变化趋势与风险值相反。风险值越高, 信任值越低。
- (2) 如果用户的风险值高于风险阈值, 那么随着风险值的增加, 用户的信任值下降的速度会越来越快。
- (3) 当风险值高于风险阈值时, 风险值的降低不会导致其信任度的增加, 只有在风险阈值高于风险值后, 风险值的降低才会导致信任度的反弹。
- (4) 用户信任值的建立是一个缓慢的过程, 需要很长时间。

基于用户的信任值和风险值, 便可较为准确地反映出用户上网行为的变化程度, 从而为提高身份认证的灵活性和准确性提供了重要的保障。

4.3 区块链安全存储与权限控制

为了提供可靠的风险评估和权限控制过程, 以及安全存储用户访问权限, 在本方案中, 利用区块链作为可信的第三方数据库。此外, 利用智能合约, 根据用户的风险值和信任值更新用户的访问权限。用户信息可以通过区块链提供的接口进行查询。此外, 用户的真实姓名将被化名取代, 以保护其隐私和匿名性。

Hyperledger 使用多通道 (Multi-channel)^[51] 的设计方案, 能够有效隔离位于不同通信信道的节点。与其他区块链平台比起来, 使用 Hyperledger 能够更加有效地保护信息的安全。更重要的是, 在 Hyperledger 平台上可以运行多种类型的智能合约, 具有较强的通用性。因此, 本文使用 Hyperledger Fabric 联盟区块链平台作为权限控制的平台。

4.3.1 安全存储与匿名保护

区块链技术能够在无需第三方参与的情况下, 采用共识机制和加密算法实现节点间相互信任, 其本质是一种去中心化的分布式数据库。此外, 共识机制和加密算法是区块链确保数据隐私与安全的有力保障。将它作为用户风险值存储的数据库, 能够有效保障用户信息的安全。

为了保护链上用户信息的匿名性, 本文链上存储与用户权限有关的相关信息时, 使用该用户的代号。本文中, 用户的代号定义为 ID^* , 它由用户 ID 加密得到, 并由服务器签名。代号 ID^* 与用户 ID 具有一一对应关系。

在从区块链中读取用户信息时, 由于读取到的是 ID^* 所标识的内容, 为了与用户 ID 对应起来, 一般情况下, 需要对 ID^* 进行解密。但是在本文中, ID^* 的作用仅为保护用户隐私, 因此不需要进行解密。为了提高整个执行过程的效率, 在验证用户代号与其真实姓名的一致性时, 不需要对用户代号进行解密, 而只需要执行签名一致性算法^[52]。定义该算法的结果为 Φ , 如公式 (4-5) 所示:

$$\Phi = (ID, ID^*, \theta) \quad (4-5)$$

其中, θ 为服务器对 ID 的签名。根据该算法验证 Φ 的值, 若 $\Phi=1$, 则说明 ID^* 是由 ID 加密并签名后得出的, 即 ID 与 ID^* 是一一对应的。若 $\Phi \neq 1$, 则表明 ID^* 不是 ID 所对应的, 则验证失败, 查询数据无效。该算法能够验证代号与用户真实姓名的一致性, 且不需要太多时间, 有效提高了认证的效率。

4.3.2 智能合约权限控制

在 Fabric 中智能合约被叫做 Chaincode, 中文翻译成链码。Chaincode 由代码和管理命令组成。其中, 代码负责主要业务的设计, 包括业务的具体执行顺序及相关逻辑。Chaincode 的部署由管理命令负责, 此外管理命令还负责保证链码的正常运行。Docker 是 Chaincode 运行的主要容器。Docker 具有轻量级及占用资源较少的特点, 能够保障 Chaincode 的运行不会占用过多资源。

在本文中, 首先利用 Fabric 的智能合约 Chaincode, 根据上一节提出的算法, 对用户进行风险评估, 然后, Chaincode 将根据用户在风险评估阶段得出的风险

值和信任值,判断用户的访问权限,最后,访问权限及用户相关信息将被存入区块链中,作为下次身份认证的凭据。本阶段的主要执行过程如下:

(1)执行 Chaincode 中的风险评估函数,得到用户的风险值 F 和信任值 Tr 。设置风险值的阈值 δ_f , 数字 1 和 0 分别表示风险程度的正常和异常情况。 R_f 是判断用户风险程度的结果,则 R_f 的取值可由式 (4-6) 表示:

$$R_f = \begin{cases} 0, & F \geq \delta_f \\ 1, & F < \delta_f \end{cases} \quad (4-6)$$

用户的信任程度可以由信任区间 ω 表示,当用户信任值在信任区间内时,可视为用户是受信任的。数字 1 和 0 分别表示用户受信任程度的正常和异常情况。 R_{Tr} 表示判断该用户是否受信任的结果。则 R_{Tr} 的取值可由式 (4-7) 表示:

$$R_{Tr} = \begin{cases} 1, & D_t \in \omega \\ 0, & D_t \notin \omega \end{cases} \quad (4-7)$$

(2) 基于风险值和信任值的评估情况,可以得到用户访问权限结果 A , 基于用该权限, Chaincode 执行权限更新函数,更新用户访问权限及相关信息,并存储到区块链中,为用户下次认证做准备。更新权限的伪代码描述如图 4-2 所示:

输入: R_f 风险程度; R_{Tr} 信任程度; U_{ID*} 标示用户的 ID

输出: NULL

ENUM A_ENUM = {suc, fal};

IF ($R_f == 1$ AND $R_{Tr} == 1$)

 A = A_ENUM.suc

ELSE

 A = A_ENUM.fal

END IF

TRY

 IF Get("/isExistUser", U_{ID*})

 A_etc = Get("/getLoginAuthority", U_{ID*})

 A_package = package(A, A_etc)

ELSE

 A_etc = A

END IF

 Send("/AuthorityUpdate", U_{ID*} , A)

CATCH Exception

 PRINT("RPC error.")

图 4-2 用户权限更新

Figure 4-2 User Authority Update

(3) 当系统下一次接收到用户的访问请求时, 若密码认证方法通过, 则将基于用户相关信息执行权限查询函数, 并通过 Fabric 区块链平台提供的智能合约接口返回查询信息, 以用户代名的形式, 返回用户访问权限结果。

查询用户访问权限的伪代码描述如图 4-3 所示:

```

输入:  $U_{ID}$ * 用户 ID
输出: A
ENUM A_ENUM = {sus, fal};
A = A_ENUM.fal
TRY
    IF Get("/isExistUser",  $U_{ID}$ *)
        A = Get("/getLoginAuthority",  $U_{ID}$ *)
    END IF
    RETURN A
CATCH Exception
    PRINT("PRC error.")
    RETURN A_ENUM.fal

```

图 4-3 用户权限查询

Figure 4-3 User Authority Query

4.4 区块链应用合理性分析

以下将通过两部分对区块链进行分析, 首先, 通过分析论证, 证明利用区块链技术能够实现用户信息的安全存储; 然后分析了利用智能合约实现用户权限控制的优势, 说明智能合约在提高权限控制安全性中的作用, 证明了将区块链技术应用在权限控制阶段中的优点与合理性。

4.4.1 区块链安全存储

本文将区块链作为安全存储用户相关风险信息数据库。区块链具有去中心化、高安全性、高冗余存储的特点, 当数据被写入区块链后, 区块链能够保障数据的安全存储, 不可伪造且不可篡改。

本文从以下两个方面证明区块链能够实现用户信息的安全存储。

证明 1: 区块链上数据在容错范围内不能被伪造和篡改。

在 Hyperledger Fabric 中, 采用 PBFT 共识算法^[53]完成数据区块的共识。设系统所需的区块链有 n 个节点, 且这些节点遵守共识机制, 由 Fabric 的共识算法可得出, 系统容错度为:

$$f = \frac{n-1}{3} \quad (4-8)$$

其中, f 代表恶意节点个数, $n = |R_j|$ 代表遵守共识机制的节点数。

接下来对共识网络的安全性进行证明。设区块链网络中的节点由以下部分组成: 共识节点, 可信节点, 恶意节点。用 N 表示所有节点, 用 $N1, N2$ 表示受信任的节点。 N' 表示恶意节点, 网络中所有的恶意节点已组成共谋。则有:

$$N = N1 \cup N2 \cup N' \quad (4-9)$$

$$N1 \cap N2 = \emptyset$$

$$N2 \cap N' = \emptyset$$

$$N1 \cap N' = \emptyset$$

当所有的节点都位于形成统一共识的过程时, 恶意节点并不能伪造新的区块。这是因为每个节点都会对自己广播过的消息进行签名。因此, 如果恶意节点想要篡改区块链内容, 就必须想办法使得系统回退到上一状态, 让区块链生成分支, 伪造出新的链。

在系统中, 如果恶意节点 N' 试图生成区块链的分支, 则其首先需要与网络中一组受信任的节点产生共识, 建立新的区块, 接着, 恶意节点要与另一组可信节点建立共识, 撤除之前和第一组可信节点产生的共识。设一开始达成共识的节点为 $N1$ 与 N' , 则如果要达成此过程, 需满足:

$$|N1| + |N'| \geq n - f, \text{ and } |N2| + |N'| \geq n - f \quad (4-10)$$

考虑最糟糕的状况, 即系统中可信节点的数目减少到了系统容忍度的临界值。此时, 有 $|N'| = f$ 。则公式 (4-10) 可化简为:

$$|N1| \geq n - 2f, \text{ and } |N2| \geq n - 2f \quad (4-11)$$

上述不等式相加得:

$$|N1| + |N2| \geq 2n - 4f \quad (4-12)$$

经化简后, 可得 $n \leq 3f$ 。又因为 $f = \frac{n-1}{3}$, 与结论不符, 则前文中假设区块链上数据在容错范围内不能被伪造和篡改的命题是成立的。

证明 2: 对于区块链节点的攻击 (如 DDOS) 将很难展开。

对于区块链来说, 区块中存储的是加密的数据, 只有拥有用户私钥的节点才能获取区块链中的核心数据, 进而获取区块内容。其次, 区块链的链式结构具有分散性、去中心化以及高冗余性的特点, 单个节点的攻破不会影响整个区块链系统的安全。最后, 区块链具有共识机制的强大算力, 在一定程度上保障了区块链的安全。因此, 攻击区块链是十分困难的。

因此,区块链能够实现用户信息的安全存储。在本文中,可将它作为安全存储用户风险控制相关数据的数据库。

4.4.2 智能合约权限控制的优越性

由于智能合约是在区块链中被执行的,因此有分布式系统的优点。本文将用户的权限控制阶段交给区块链智能合约来控制,如图 4-4 所示,在权限控制过程中,使用智能合约具有以下优越性:

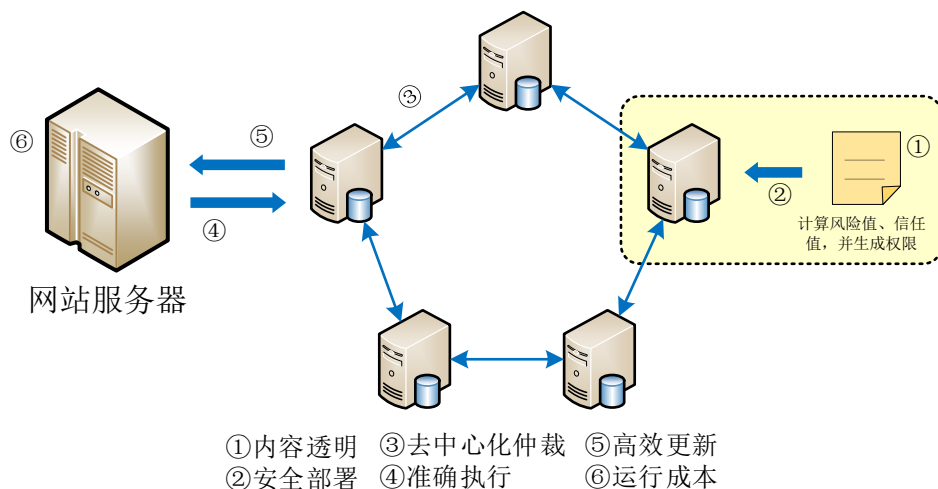


图 4-4 利用智能合约控制权限的优越性
Figure 4-4 Advantages of Smart Contract Authority Controlling

(1) 执行权限控制的去中心化仲裁。在本文的方案中,权限控制交于智能合约,权限控制的过程是否执行不需要中心化的权威机构来决定,而是由区块链网络中绝大部分的节点通过共识机制来决定执行的,使得身份认证的过程更加透明化且具有信服力。

(2) 智能合约的安全部署。在本文提出的方案中,一旦负责权限控制的智能合约部署在区块链上,合约的所有内容都将无法修改。即一旦权限控制智能合约生效,其风险评估及权限控制的方法将是确定且不可篡改的。因此,攻击者无法通过篡改合约内容进行攻击。

(3) 智能合约内部权限控制方法的透明化。由于智能合约是基于区块链的,因此部署在区块链上的智能合约内容也是公开透明的,因此,只要权限控制方法能够得到公众的信任,则基于对代码的信任,就可以在各种环境下安心、安全地进行权限控制。

(4) 权限控制执行结果的准确性。一旦相关条件被触发,计算机将自动执行制定好的合约内容。因此,在保证权限控制算法正确性的情况下,能够确保得到无差错的用户权限,从而保证了权限查询结果的准确性。

(5) 用户访问权限更新的实时性。只要满足触发条件, 智能合约就能够自动执行, 而无需等待其他中间机构。用户权限值 A 不需要等待第三方机构的参与, 而是可以通过智能合约自动更新, 从而, 能够保障系统查询到的是用户的最新权限。通过智能合约进行权限控制, 得到的权限结果更具有实时性和准确性。

(6) 节约运行成本。智能合约的运行不需要人的参与, 在合约的执行与裁决等过程中都能够达到节约成本的效果。利用智能合约控制用户的权限, 可以大大节省在合约执行期间的人力, 从而降低整个身份认证过程的运行成本。

总之, 智能合约作为区块链中一种重要的工具, 具有去信任, 无需第三方仲裁, 经济高效的特点。通过智能合约控制用户访问权限, 能够保障权限更新与查询过程的安全性和准确性, 在保证经济效益的同时提高身份认证的效率。

4.5 本章小结

本章主要叙述了基于行为的风险评估及权限控制方法。首先, 描述了利用 Session 来记录用户上网行为的方法。利用 Session ID 标记用户每次的上网行为, 可以较为直观且准确地记录用户出现的相关操作, 为后期风险评估做准备。其次, 介绍了风险评估的方法, 用户的风险行为利用风险值和信任值来描述, 既能够反映用户在短期内风险的变化, 又能够反映该用户受信任的程度, 使得风险评估结果更具有灵活性和准确性。最后, 将区块链用于用户信息的安全存储以及权限控制。区块链是分布式的可信数据库, 利用区块链存储用户相关信息, 可以有效地保障用户相关信息的安全性以及可追溯性。另外, 基于 Hyperledger Fabric 的智能合约 Chaincode 对用户权限进行控制, 能够使得权限控制更加智能化, 有效提高了身份认证的安全性。

第5章 系统实现及实验结果分析

本章根据第三章与第四章提出的身份认证方法与风险评估方案,首先,以校园网作为落地场景,完成了基于风险的身份认证系统的实现。其次,测试了当用户无风险操作与出现风险操作后系统的处理情况,完成了系统的功能性验证。另外,以 Hyperledger Fabric 为平台,进行了智能合约的编写,并通过接口验证了合约内容的有效性。最后,通过实验验证了系统的效率。实验证明,系统能够在效率损失合理的范围内,有效提高身份认证的安全性与灵活性。

5.1 身份认证系统开发

本节描述了基于 UCON 模型的动态身份认证系统的落地实现。基于第三章和第四章提出的身份认证与权限控制方法,对整个系统进行了较为完整的实现。下文将对开发环境,系统实现等相关内容进行介绍。

5.1.1 环境配置

表 5-1 和表 5-2 描述了系统工作的软硬件配置。

表 5-1 系统工作硬件配置
Table 5-1 System Working Hardware Configuration

配置	描述
操作系统	Ubuntu16.04
CPU	Intel (R) Core (TM) i5-4200M
内存	6GB

表 5-2 系统工作软件配置
Table 5-2 Table 5-1 System Working Software Configuration

开发环境名称	描述
Linux	服务器环境
Goland	开发工具
Golang	编程语言
B/S	开发模式
Golang1.10	开发语言版本

系统的实现基于上文提到的基于 UCON 模型的动态风险身份认证方法。网站用户是系统的主要服务对象。当登录请求被发送给系统，系统首先查询是否有该用户的注册信息，如果没有，则跳转到注册阶段，若有，则跳转到身份认证阶段。最后，用户接收系统认证结果。具体步骤如下：

(1) 用户身份查询 对用户信息进行查询，若查询不到该用户，则进入注册阶段，若能，则进入身份认证阶段。

(2) 用户信息注册 对用户相关信息进行注册，并将用户初始权限发送到区块链中进行存储。

(3) 身份认证 根据用户密码进行密码认证，并从区块链查询用户权限信息。若用户通过密码认证且具有访问权限，则该用户通过认证。

(4) 查询并返回结果 身份认证结果将发送给用户。

5.1.2 系统实现

上文已经详细介绍了身份认证系统的工作过程。系统采用 B/S 的交互模式，使用 Chrome 等网页浏览器进行交互。图 5-1 是系统登录界面，用户需要输入用户名和密码进行登录，以便系统进行后续操作。

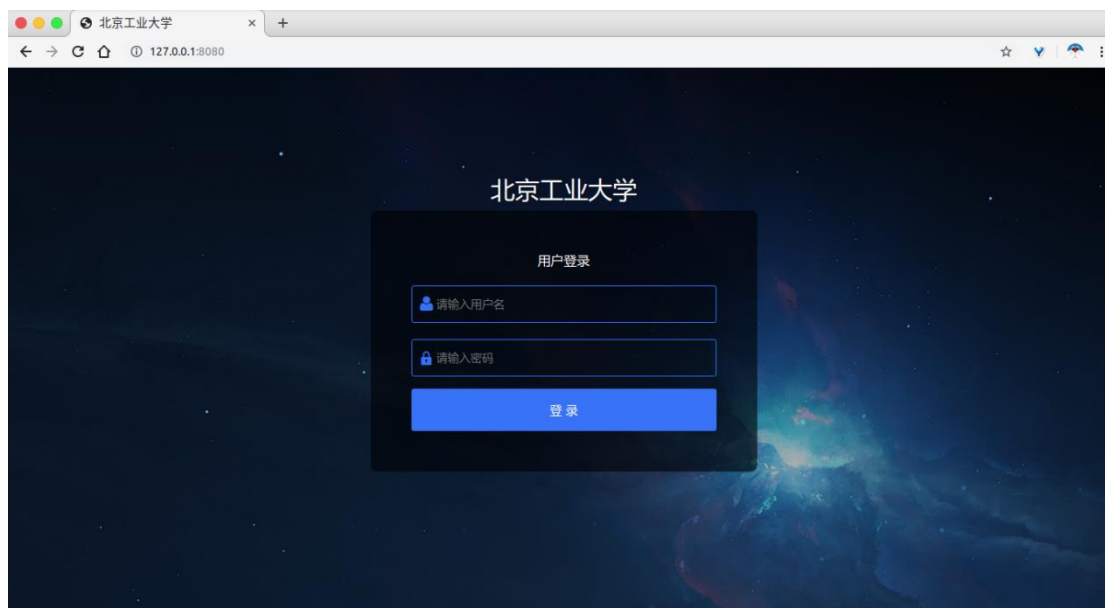


图 5-1 登录界面

Figure 5-1 Login Page

图 5-2 是用户登录成功界面，若用户登录成功，则显示校园网主页，用户可进行对校园网站内相关资源的访问。



图 5-2 登录成功界面

Figure 5-2 Login Success Page

图 5-3.图 5-4 描述了网站子界面，用户可在左侧页面中进行相关操作。如查询学生信息，修改学生信息，选课等。

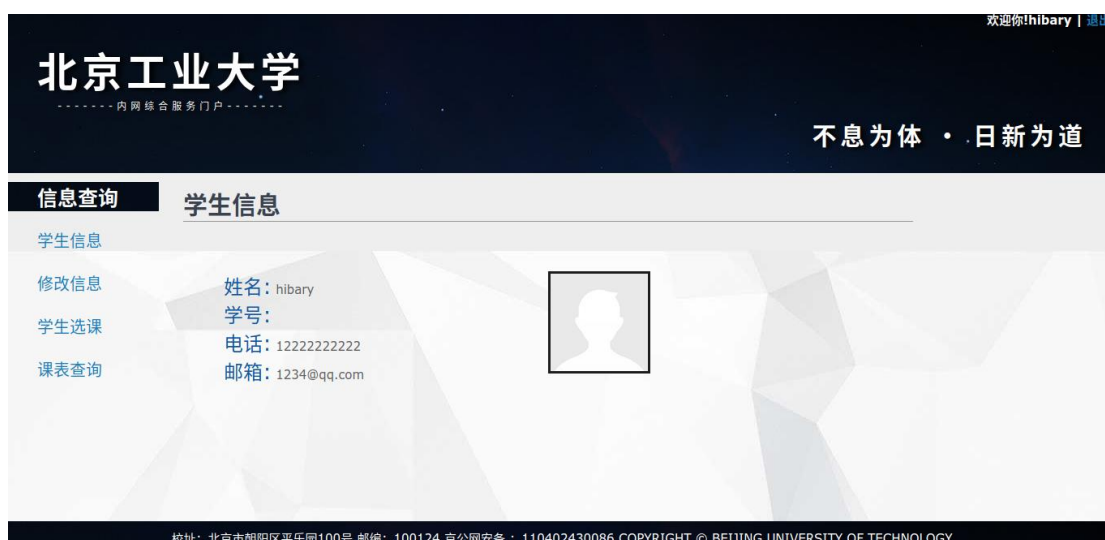


图 5-3 学生信息子页面

Figure 5-3 Student Information Sub Page



图 5-4 学生信息修改子界面

Figure 5-4 Student Information Modification Sub Page

以上界面描述了当用户登录成功时系统返回的结果。同时，还存在着用户无法正常登录的情况。如下图所示。

场景 1: 若用户名密码输入错误，则系统返回用户名或密码错误提示。如图 5-5 所示。

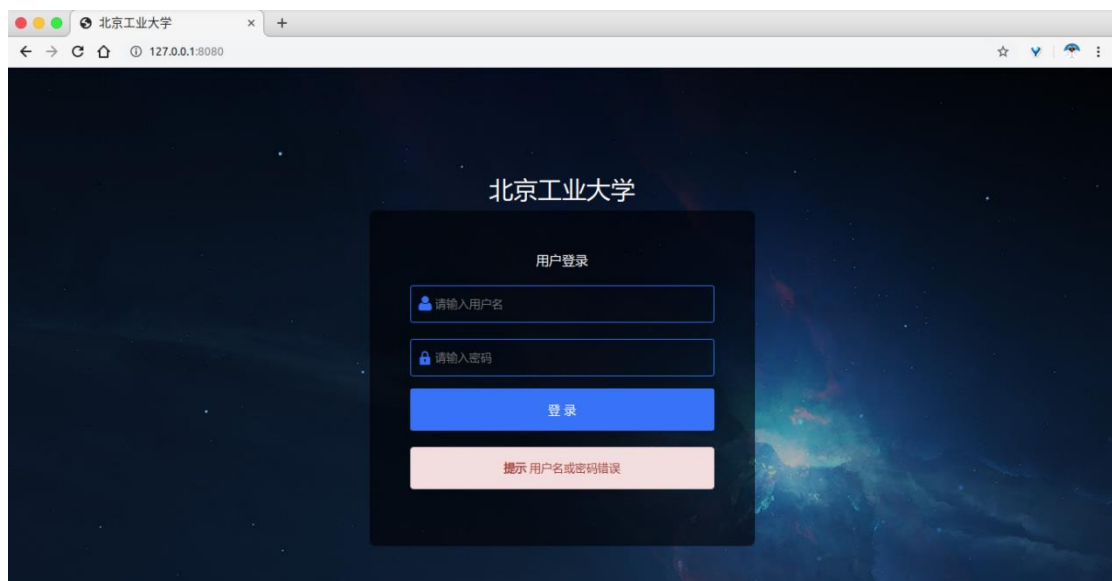


图 5-5 用户名密码错误界面

Figure 5-5 User Name and Password Error Page

场景 2: 若用户通过了密码认证，但由于用户风险过高，也将返回拒绝登录提示，如图 5-6 所示。

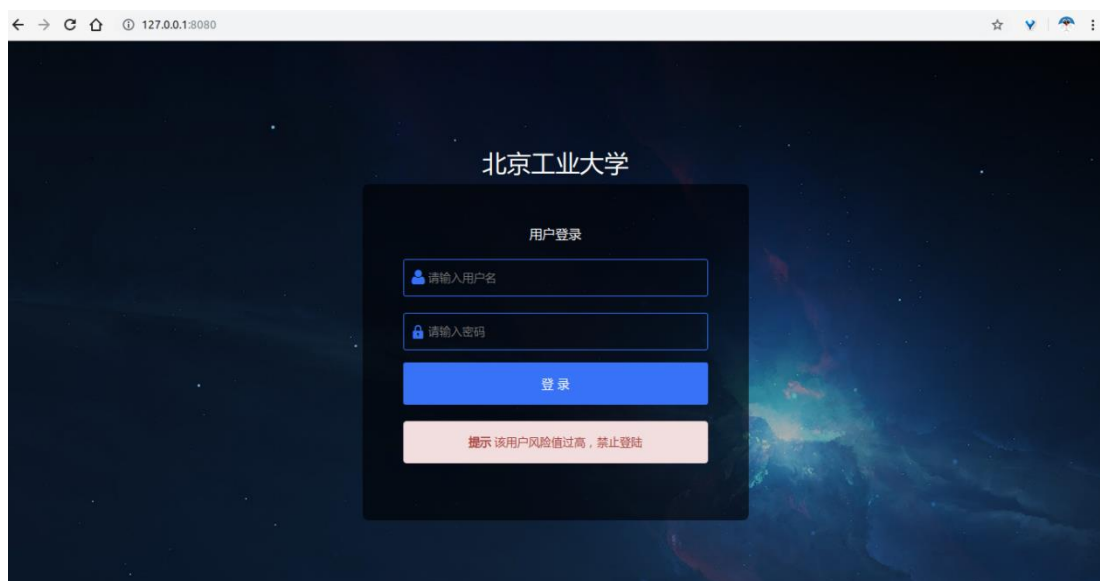


图 5-6 风险过高界面

Figure 5-6 High Risk Page

由此可见, 本认证系统实现了基于风险的身份认证方法的基本功能, 当用户请求登录时, 该系统能根据用户的风险有效控制用户的访问权限, 能够有效保护网站资源的安全, 提高身份认证的安全性及灵活性。

5.2 区块链平台相关设计与实现

本节主要叙述了基于Hyperledger Fabric 联盟区块链平台的相关工作。由于本文提出的方案应用在校园网等中小型网络中, 其物理网络组成如表 5-3 所示:

表 5-3 Fabric 物理网络组成

Table 5-3 Fabric physical network composition

名称	描述
Fabric CA	整个基本网络的证书颁发机构 (CA), 生成客户端认证
Client	客户端, 提供向服务器查询的接口
Peer0,1,2	网络中的节点

5.2.1 认证系统与 Fabric 衔接

为了实现认证系统与 Hyperledger Fabric 的衔接, 本文在服务器与 Fabric 区块链平台之间设立了 RPC 模块, 负责服务器与 Fabric 之间的通信。该模块使用 node.js 开发, 具有便于开发与配置的特点。

在启动本系统时共需要启动 3 种模块，包含启动 Fabric 模块、RPC 模块和网站后端模块，启动管理脚本分别是 fabric.sh、rpc-server.sh 和 BjutLgn。首次需要创建 fabric 网络，使用 fabric.sh 的 create 参数创建，会根据配置文件创建对应的网络，之后使用 fabric.sh 文件的 start 参数正常启动 fabric 网络；fabric.sh 的 start-rpc 参数启动可以启动 rpc 服务，两个依赖服务启动后，后端可以通过 BjutLgn 打包文件正常启动了。

整个系统的启动过程如图 5-7 所示：

```
hibary@hibary-vm ~/BjutAuth> ls
fabric/ install.sh* readme.MD* run-create.sh* run.sh* source/ stop.sh*
hibary@hibary-vm ~/BjutAuth> ./run.sh
启动fabric网络...
启动fabric网络完毕!
Starting ca.example.com ... done
Starting orderer.example.com ... done
Starting couchdb ... done
Starting peer0.org1.example.com ... done
Starting cli ... done
启动rpc服务...
启动rpc服务完毕!
nohup: redirecting stderr to stdout
启动网站主程序...
网站主程序启动完毕!
hibary@hibary-vm ~/BjutAuth>
```

图 5-7 系统启动界面
Figure 5-7 System Startup Page

5.2.2 基于 Fabric 的智能合约设计

Chaincode 主要使用 golang 语言开发，它是在结构网络内部执行的（在支持对等体上）。它是所有参与者都同意的业务逻辑所在的位置，并且在满足条件时执行。链码遵循 Hyperledger Fabric 中定义的特定模式。基于智能合约实现的主要功能，设计的接口如表 5-4 所示：

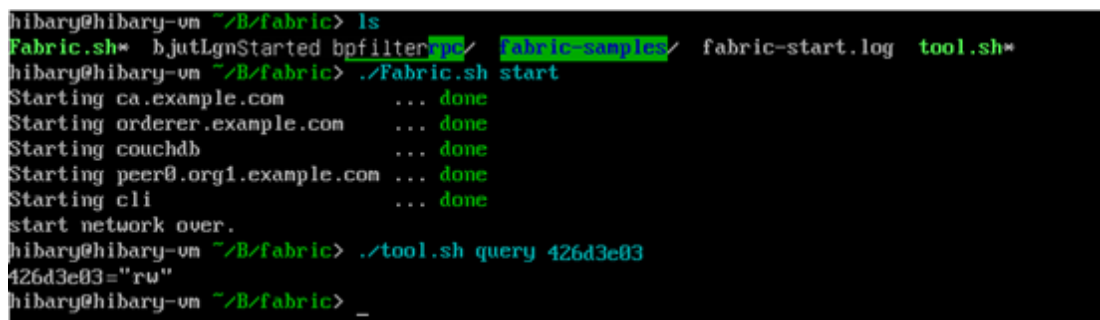
表 5-4 智能合约接口
Table 5-4 smart contract interface

接口名称	接口说明
/initUserAuthority	初始权限注册
/getLoginAuthority	登录权限查询
/getUserAuthorityBySessionID	根据 SessionID 查询风险行为
/isExistUser	验证用户信息是否存储在区块链中
/AuthorityuUpdate	权限更新
/saveUserAuthorityBySessionID	根据 SessionID 存储风险行为

以学生身份认证功能为例进行说明智能合约函数之间的调用。首先系统调用/isExistUser 函数判断该用户是否存在，如果不存在则报错，并准备调用/initUserAuthority 为该生进行注册，若存在，系统调用/getLoginAuthority，查询该学生访问权限，返回访问结果。

5.2.3 智能合约功能性验证

如图，可利用 Fabric 提供的接口验证智能合约的功能。如图 5-8，在界面中输入用户 ID 并调用相关指令，可查询到用户（此处为代号）的权限值为可读写。



```
hibary@hibary-vm ~/B/fabric> ls
Fabric.sh*  bjutLgnStarted bpfilterrpc/ fabric-samples/ fabric-start.log  tool.sh*
hibary@hibary-vm ~/B/fabric> ./Fabric.sh start
Starting ca.example.com      ... done
Starting orderer.example.com ... done
Starting couchdb             ... done
Starting peer0.org1.example.com ... done
Starting cli                  ... done
start network over.
hibary@hibary-vm ~/B/fabric> ./tool.sh query 426d3e03
426d3e03="rw"
hibary@hibary-vm ~/B/fabric> _
```

图 5-8 智能合约查询 1

Figure 5-8 Smart Contract Query 1

在主页面登录该用户的账户，并进行一些风险操作。然后，再次在界面中调用相关指令，如图 5-9 所示，可查询到用户（此处为代号）的权限值变为了只读。



```
hibary@hibary-vm ~/B/fabric> ls
Fabric.sh*  bjutLgnStarted bpfilterrpc/ fabric-samples/ fabric-start.log  tool.sh*
hibary@hibary-vm ~/B/fabric> ./Fabric.sh start
Starting ca.example.com      ... done
Starting orderer.example.com ... done
Starting couchdb             ... done
Starting peer0.org1.example.com ... done
Starting cli                  ... done
start network over.
hibary@hibary-vm ~/B/fabric> ./tool.sh query 426d3e03
426d3e03="rw"
hibary@hibary-vm ~/B/fabric> ./tool.sh query 426d3e03
426d3e03="r"
hibary@hibary-vm ~/B/fabric>
```

图 5-9 智能合约查询 2

Figure 5-9 Smart Contract Query 2

并且，可以根据用户本次上网的 Session id，查询到用户本次访问过程中的相关风险行为，如图 5-10。


```
hibary@hibary-vm ~/B/fabric> ls
Fabric.sh*  bjutLgnStarted bpfilter-pe/ fabric-samples/  fabric-start.log  tool.sh*
hibary@hibary-vm ~/B/fabric> ./Fabric.sh start
Starting ca.example.com      ... done
Starting orderer.example.com ... done
Starting couchdb             ... done
Starting peer0.org1.example.com ... done
Starting cli                  ... done
start network over.
hibary@hibary-vm ~/B/fabric> ./tool.sh query 426d3e03
426d3e03="suc"
hibary@hibary-vm ~/B/fabric> ./tool.sh query 426d3e03
426d3e03="fal"
hibary@hibary-vm ~/B/fabric> ./tool.sh querySession 89b40f50-1872-4d82-a45d-6b416bb18751
{"session":"89b40f50-1872-4d82-a45d-6b416bb18751", "url":"/homepage", "actionType":"locale of login
error", "time":"1581879312"}
{"session":"89b40f50-1872-4d82-a45d-6b416bb18751", "url":"/ChangeInfo", "actionType":"exceeds author
ized access", "time":"1581879808"}
{"session":"89b40f50-1872-4d82-a45d-6b416bb18751", "url":"/Information", "actionType":"exceeds autho
rized access", "time":"1581880782"}
hibary@hibary-vm ~/B/fabric>
```

图 5-10 风险行为查询
Figure 5-10 Risk Behavior Query

由此可见，借助智能合约，可以实现用户权限的自动控制，有效地提高了身份认证的安全性。

5.3 实验结果及分析

5.3.1 实验规模及场景设置

这里主要考虑用户身份认证过程的计算时间，因为该阶段是本方案的主要执行部分，并且比其他阶段更加频繁地执行。考虑以下场景：某校园网的用户接入过程。区块链平台选取 Hyperledger Fabric，由校园网服务器执行接入认证。考虑到选取网络的规模较小，本方案选取了 1、10、50、100 个用户的情况，分别在这些用户同时接入的场景下，分别进行了 100 次测试。使用如下配置的设备模拟校园网服务器，如表 5-5 所示：

表 5-5 实验环境
Table 5-5 Experimental Environment

配置	描述
带宽	100Mbps
CPU	Intel (R) Core (TM) i5-4200M
内存	6GB

5.3.2 实验设计及结果分析

本节共设计了三个实验，分别从效率，稳定性及区块链性能方面测试了本系统的相关性能。

实验 1：系统稳定性测试

本文中选取了校园网作为登录场景,本实验中,分别模拟了1、10、50、100名用户同时请求登录的时候,系统对每位用户的响应时间。对以上每种同时请求登录的情况均进行了100次实验。分析比较结果,如图5-11所示:

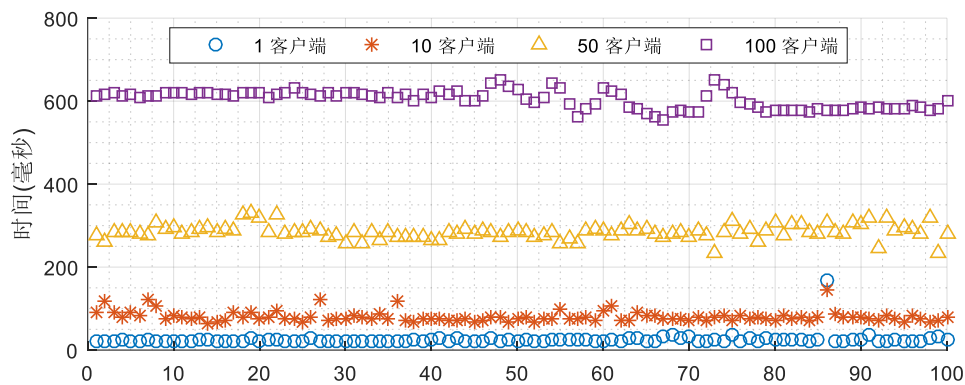


图 5-11 用户接入时间
Figure 5-11 User Access Time

图 5-11 反映了这些用户的平均接入延迟。从该图中可以看出,当多个用户同时接入时,系统对每个用户的响应时间都是差不多的,没有出现时间严重不均的情况。由此可见,本文搭建的系统具有较强的稳定性,在同时接入多个用户时,仍然可以正常运作。

实验 2: 系统登录效率测试

在本实验中,对某位用户来说,分别模拟了当该用户与不同数目用户同时请求登录时,系统对该用户登录的响应时间。并对响应变化进行了记录。

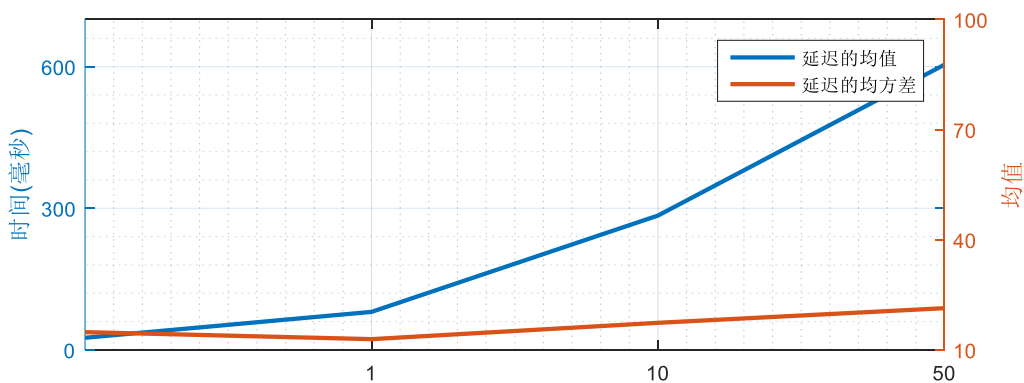


图 5-12 单名用户接入时间
Figure 5-12 Access Time of Single User

图 5-12 反映了该用户接入实验的均值和方差，能够反映单个用户接入所需响应时间的变化趋势。可以看出，即使在多个用户同时登录的情况下，本方案效率受到的影响仍保持在可以接受的范围。

实验 3：区块链性能测试

本节选取 100 名用户同时接入的场景，对于本系统在使用区块链进行权限控制与不使用区块链进行权限控制的情况下，对于用户的接入时间进行了对比。

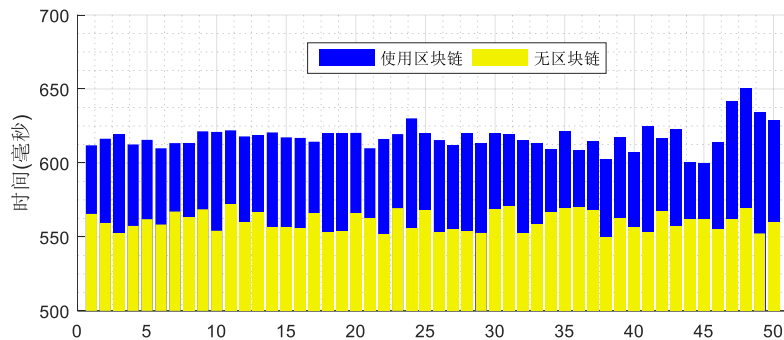


图 5-13 两种情况下系统响应时间

Figure 5-13 System Response Time in Two Cases

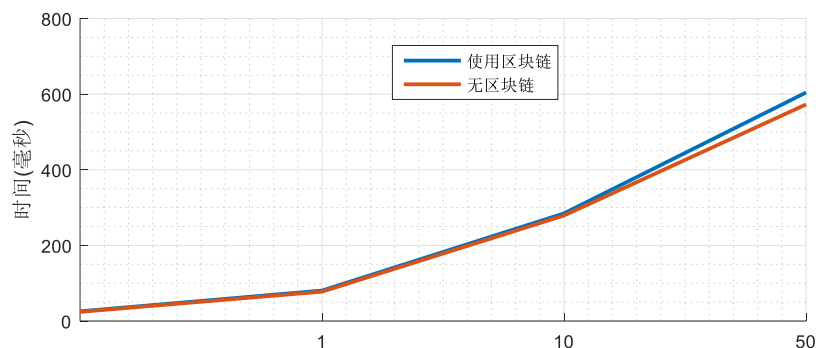


图 5-14 用户接入时间变化

Figure 5-14 Change of user access time

图 5-13 反映了用户接入时，使用区块链与不使用区块链时，系统对用户的响应时间。图 5-14 反映了用户在两种情况下，同时接入的用户数增加时，系统对用户的响应时间变化趋势。可以看到，区块链对于用户接入时间的影响保持在可以容忍的范围，并且，接入区块链不是影响系统性能的决定性因素。尤其是，由于本方案的重点是保障用户能够安全接入，因此对安全性的要求相对较高，因此相应地会牺牲一些性能。

此外,方案仍对工作效率进行了优化,例如,利用签名一致性算法代替解密用户代名的方法以提高检索效率。在克服了一些安全缺陷的同时,在一定程度上提高了效率。

5.4 本章小结

本章根据第三章第四章提出的框架,完成了基于 UCON 模型的动态风险身份认证系统的实现。首先,介绍了身份认证系统的开发环境及系统的整体使用流程,并利用 Goland 等工具完成了身份认证系统的整体开发,并展示了系统的功能。然后,基于 Hyperledger Fabric 区块链平台进行了智能合约的开发与部署,实验结果同样验证了智能合约有效性。最后,在一定的实验规模及场景设置下,对整个系统的效率及性能进行了分析。实验证明,即使在多个用户同时登录的情况下,本方案效率受到的影响仍保持在可以接受的范围内。

结 论

当今时代,复杂多变的网络环境对身份认证的安全性和灵活性提出了更高的要求。对于现存的身份认证方法来说,连续性与灵活性的缺乏是其存在的主要问题。虽然 UCON 模型可以在一定程度上满足当前网络的需求,但其仍存在一些缺陷。另外,区块链作为一种去中心化的分布式数据库,其链式结构、共识机制等特性能够有效保障网络环境的安全性,在身份认证领域的应用也愈发广泛。因此,本文基于 UCON 模型提出了一种基于风险的动态身份认证方法。本文将风险评估、权限控制与身份认证相结合,并将区块链技术应用于为风险评估和权限控制中。

基于以上问题,首先,本文提出了基于 UCON 模型的动态风险身份认证方法。该方法在对用户进行认证时,将参考密码认证与权限控制的共同结果。用户不仅需要通过密码认证,还需要具有相应的访问权限,只有这样用户才能正常地访问网络资源。其次,本文提出了基于用户历史行为的风险评估的方法。当用户退出访问后,将对用户本次的访问行为进行分析和风险评估,并得出用户的风险值和信任值,以作为控制用户的访问权限的依据。最后,本文利用区块链作为可信的第三方数据库,并利用智能合约,根据用户的风险值和信任值更新用户的访问权限。经论证,方案提出的动态身份认证方法能够有效提高身份认证的灵活性,用户行为采集及风险评估提高了身份认证的连续性,而区块链的使用实现了在去中心化的网络环境中对用户权限进行控制,从而能够提高身份认证的安全性。本文提出了方案改进了传统的身份认证方法,能够有效保障网络环境的安全。

本文基于提出的方案,对认证方案的安全性进行了理论分析。本文使用 SVO 逻辑对方案进行逻辑推理,并在多种攻击场景下对本方案进行分析,表明本方案是安全可行的,并可以抵御多种网络环境中的攻击。本文以校园网作为落地场景,在理论的基础上完成了系统的开发与实现,并使用 Hyperledger Fabric 作为区块链平台,进行了智能合约的编写与部署。实验证明,该系统能够基于用户的风险行为很好地实现用户权限的控制,在花费了较低开销的同时具备了较好的效率和性能。

本文的方案改进了传统的身份认证方法,具有较高的灵活性,连续性和安全性。经证明,基于本方案的认证系统具有较高的稳定性和效率。此外,本方案具有较强的可扩展性,可应用于各种需要对用户进行身份认证的场景。本文针对基于风险的身份认证与区块链相结合的研究是一个初步的研究,在未来希望能有更多的认证方法和策略。

参 考 文 献

- [1] Internet World Stats. Internet usage statistics; The Internet big picture: World Internet users and 2019 population stats [EB/OL]. (2019-3-13)[2019-3-19]. <https://www.internetworldstats.com/stats.htm>.
- [2] Widup S, Spitler M, Hylender D, et al. Verizon data breach investigations report [R]. Tech Rep. 2018-4.
- [3] The Internet Engineering Task Force. Internet Security Glossary[EB/OL]. (2013-03-02)[2019-8-22]. <https://datatracker.ietf.org/doc/rfc2828>.
- [4] Mirian A, DeBlasio J, Savage S, et al. Hack for hire: Exploring the emerging market for account hijacking [EB/OL]. (2019-3-13)[2020-2-12]. <https://www.sysnet.ucsd.edu/~voelker/pubs/hackforhire-www19.pdf>.
- [5] Akamai Technologies. Soti-2018-credential-stuffing-attacks-report [DB/OL]. (2019-3-13)[2020-2-13]. <https://www.akamai.com/cn/zh/multimedia/documents/state-of-the-internet/soti-2018-credential-stuffing-attacks-report.pdf>.
- [6] Patil P, Zavarsky P, Lindskog D, Ruhl R. Fault tree analysis of accidental insider security events [C]//International Conference on Cyber Security (ICCS). IEEE, Washington D.C., USA, 2012: 113–118.
- [7] 崔久强, 徐祺. 移动互联网身份认证技术研究[J]. 信息安全与技术. 2015(07)
- [8] Yang F Y, Hsu C W, Chiu S H. Password authentication scheme preserving identity privacy [C]//Sixth International Conference on Measuring Technology and Mechatronics Automation (SICMTMA). IEEE, Washington D.C., USA, 2014: 443–447.
- [9] Song R. Advanced smart card based password authentication protocol [J]. Computer Standards & Interfaces. 2010, 32(5-6): 321–325.
- [10] Jangirala S, Mukhopadhyay S, Das A K. A multi-server environment with secure and efficient remote user authentication scheme based on dynamic ID using smart cards [J]. Wireless Personal Communications. 2017, 95(3): 2735–2767.
- [11] Odelu V, Das A K, Goswami A. A secure biometrics-based multi-server authentication protocol using smart cards [J]. IEEE Transactions on Information Forensics and Security. 2015, 10(9): 1953–1966.
- [12] He D, Wang D. Robust biometrics-based authentication scheme for multiserver environment [J]. IEEE Systems Journal. 2015, 9(3): 816–823.
- [13] Sun J, Zhong Q, Kou L, et al. A lightweight multi-factor mobile user authentication scheme [C]//IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, Washington D.C., USA, 2018: 831–836.

- [14] RSA Security. RSA risk-based authentication [EB/OL]. (2013-11-12)[2019-3-17]. http://webobjects.cdw.com/webobjects/media/pdf/rsa/H11465_RBA_WP_0113.pdf?cm_sp=RSAShowcase-_-Cat1Link4-_SecurID+White+Paper.
- [15] Luo B, Liu Y. The risk evaluation model of network information security based on improved BP neural network [C]//International Symposium on Instrumentation & Measurement, Sensor Network and Automation (IMSNA). IEEE, Washington D.C., USA, 2012: 189–191.
- [16] Patil P, Zavarsky P, Lindskog D, Ruhl R. Fault tree analysis of accidental insider security events [C]//International Conference on Cyber Security (ICCS). IEEE, Washington D.C., USA, 2012: 113–118.
- [17] Hong Q, Jianwei T, Zheng T, et al. An information security risk assessment method based on conduct effect and dynamic threat [C]. 8th IEEE International Conference on Software Engineering and Service Science (ICSESS). IEEE, Washington D.C., USA, 2017: 782–786.
- [18] Kosba A, Miller A, Shi E, Wen Z and Papamanthou C. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts [C]//IEEE Symposium on Security and Privacy (SP). IEEE, Washington D.C., USA, 2016: 839–858.
- [19] Sanda T, Inaba H. Proposal of new authentication method in Wi-Fi access using Bitcoin 2.0. Consumer Electronics[C]//2016 IEEE Global Conference on. IEEE, Washington D.C., USA, 2016: 1–5.
- [20] Raju S, Boddepalli S, Gampa S, Yan Q, Deogun J S. Identity management using blockchain for cognitive cellular networks[C]//IEEE International Conference on Communications. IEEE, Washington D.C., USA, 2017: 1–6.
- [21] Cruz JP, Kaji Y, Yanai N. RBAC-SC: Role-based access control using smart contract [J]. IEEE Access. 2018: 12240–12251.
- [22] Dorri A, Kanhere SS, Jurdak R, Gauravaram P. Blockchain for IoT security and privacy: The case study of a smart home [C]//2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). IEEE, Washington D.C., USA, 2017: 618–623.
- [23] Zyskind G, Nathan O. Decentralizing privacy: Using blockchain to protect personal data [C]//IEEE Security and Privacy Workshops. IEEE, Washington D.C., USA, 2015: 180–184.
- [24] Christidis K, Devetsikiotis M. Blockchains and smart contracts for the Internet of Things [J]. IEEE Access. 2016: 2292–2303.
- [25] Azaria A, Ekblaw A, Vieira T, et al. Medrec: Using blockchain for medical data access and permission management [C]//2nd International Conference on Open and Big Data (OBD). IEEE, Washington D.C. USA, 2016: 25–30.
- [26] Ramachandran A, Kantarcioglu D. Using blockchain and smart contracts for secure data provenance management [J]. arXiv preprint arXiv. 2017: 1709.10000.
- [27] Yuan Y, Wang F Y. Blockchain and Cryptocurrencies: Mode Techniques, and Applications[J]. IEEE TRANSACTIONS ON SYSTEMS MAN CYBERNETICS-SYSTEMS. 2018, 48(9):1421-1428.

- [28] Androulaki E, Barger A, Bortnikov V, et al. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains [J]. 2018. DOI: 10.1145/3190508.3190538.
- [29] 范建淑,张俊霞.论网络安全中的身份认证技术[J]. 网络安全技术与应用. 2018(01).
- [30] 周棟淞,杨洁,谭平嶂,庞飞,曾梦岐.身份认证技术及其发展趋势[J]. 通信技术, 2009: 183-185.
- [31] Ouda A. A framework for next generation user authentication[C]//2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC), Muscat, 2016: 1-4.
- [32] Jaehong P, Ravi S. The UCONABC usage control model[J]. ACM Transactions on Information and System Security (TISSEC). 2004.
- [33] Um-e-Ghazia, Masood R, Shibli M A, et al. Usage Control Model Specification in XACML Policy Language[C]//IFIP International Conference on Computer Information Systems and Industrial Management. Springer Berlin Heidelberg. New York, USA. 2012.
- [34] 蔡婷,陈昌志.云环境下基于 UCON 的访问控制模型研究[J]. 计算机科学, 2014: 262-264.
- [35] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system [EB/OL]. (2018-03-21) [2019-02-15]. <https://bitcoin.org/bitcoin.pdf>.
- [36] Butian Huang, Zhenguang Liu, Jianhai Chen. Behavior Pattern Clustering in Blockchain Networks[J]. Multimedia Tools and Applications, 2017, 76(19):20099-20100.
- [37] 董宁,朱轩彤.区块链技术演进及产业应用展望[J]. 信息安全研究, 2017, 3(3): 200-210.
- [38] Hyper ledger White paper. [EB/OL]. (2018-04-10)[2019-02-14]. <https://wiki.hyperledger.org/groups/whitepaper/whitepaper-wg> accessed 10 January.
- [39] 袁勇, 王飞跃. 区块链技术发展现状与展望[J]. 自动化学报, 2016, 42(04):481-494
- [40] IBM. Hyperledger Fabric Technical Document[EB/OL]. (2019-01-08)[2019-02-17]. <https://fabric.io>.
- [41] 项俊龙, 陈传峰. 安全协议形式化验证方法综述[J]. 信息安全与通信保密, 2013, (5): 5254.
- [42] Burrows M, Abadi M, Needham R. A logic of authentication [J]. Operating Systems Review, ACM SIGOPS, 1989, 23(5): 1-13.
- [43] Gong L, Needham R, Yahalom R. Reasoning about belief in cryptographic protocols [C]//1990 IEEE Computer Society Symposium on Research in Security and Privacy, IEEE, Washington D.C. USA, 1990: 234-248.
- [44] Abadi M N, Tuttle M R. A semantics for a logic of authentication [J]. Proceedings of ACM Annual Symposium on Principles of Distributed Computing, IEEE, Washington D.C. USA, 1991: 18-36.
- [45] Oorschot P V. Extending cryptographic logics of belief to key agreement protocols [C]//CCS '93, Proceedings of the ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, 1993: 232-243.
- [46] Thakur T, Dogra S, Sood Y. A Review and Comparative Analysis of Modal Logics: BAN, GYN and SVO[J]. International Journal of Research and Engineering, 2015, 2(4): 30-33.

- [47] Darwish M, Ouda A, Capretz L F. Formal Analysis of an Authentication Protocol Against External Cloud-Based Denial-of-Service (DoS) Attack[J]. arXiv preprint arXiv, 2017, 1711.09985.
- [48] Wikipedia. One-key_MAC [J/OL]. Wikipedia, 2019: 9-8[2019-02-17]. https://en.wikipedia.org/wiki/One-key_MAC.
- [49] Rughoobur P, Nagowah L. A lightweight replay attack detection framework for battery depended IoT devices designed for healthcare [C]//2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS). IEEE, Washington D.C. USA, 2017: 811–817.
- [50] Neelima G, Rodda S. Predicting user behavior through sessions using the web log mining [C]//International Conference on Advances in Human Machine Interaction (HMI). IEEE Washington D.C. USA, 2016: 1–5.
- [51] Androulaki E, Barger A, Bortnikov V, et al. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains [J]. 2018. DOI: 10.1145/3190508.3190538.
- [52] Choon J C, Cheon J H. An identity-based signature from gap Diffie-Hellman groups[C]//International workshop on public key cryptography. Springer, Berlin, Heidelberg, 2003: 18-30.
- [53] Castro M, Liskov B. Practical byzantine fault tolerance. In: Proceedings of the third Symposium on Operating Systems Design and Implementation. New Orleans, Louisiana, USA: OSDI, 1999: 173–186

攻读硕士学位期间发表的学术论文

- 1 专利: 刘静,刘荣超,赖英旭,吕建富,原昌博. 一种基于区块链技术的身份认证方法. (申请号或专利号: 201811462299.7).
- 2 论文 :Jing Liu, Rongchao Liu, and Yingxu Lai. Risk-based dynamic identity authentication method based on the UCON model. Security and Communication Networks. (在投).
- 3 计算机软件著作权: 刘荣超,刘静,赖英旭. 基于风险的动态身份认证系统. (登记号: 2019SR1141402).

致 谢

时光飞逝，研究生生活已经接近了尾声。在这三年的学习生活中，我对专业领域的知识有了更加深入的掌握。在这不算长也不算短的三年时光里，我很庆幸能够拥有一批在学习和生活中给予我指导和帮助的良好益友，正是有了你们的支持和付出，我的研究生生活才能如此丰富多彩。

首先，我要特别感谢赖英旭老师和刘静老师，犹记得当年入学的时候，自己十分迷茫和彷徨，是老师们不断帮助我，引导我，使我逐渐养成了独立思考的能力，形成了自己的学术思维体系，找到了自己的研究方向。从选方向，到开题，再到论文的写作和定稿，都离不开赖老师和刘老师的辛勤付出和悉心指导。不仅如此，在科研中老师们专注严谨的精神，以及在工作中老师们认真负责的态度，都深深地打动了我。在以后的工作中，我也会保持着这份认真负责的态度，表现出更好的自己。此外，在生活中，两位老师更是给予了我很多关怀和鼓励，使我能够一路披荆斩棘，对未来充满信心。非常感谢两位老师对我的帮助和指导，我的成长离不开老师的辛勤付出。

其次，要感谢我已经毕业的师兄师姐和我的同学们。感谢建富师兄，静雯师姐等师兄师姐对我的照顾和帮助。感谢昌博、小马、尊旭、安康以及所有信南 511 的师弟师妹们对我的帮助，和你们一起，我度过了很快乐的时光。感谢我的舍友姜倩、刘曼、付璐在生活中的陪伴，有了你们我的生活才变得更加丰富多彩。感谢张聪同学对我在学习和生活中的照顾以及找工作时的鼓励，感谢有你。

最后，我要感谢我的家人。感谢父母对我的养育之恩，感谢你们这么多年对我的付出和关怀。感谢姥姥姥爷对我的关心和疼爱。无论到哪里，你们都是我最温馨的港湾。

在这里，我再次由衷的感谢所有在我遇到困难时给予我帮助的人。三年的沉淀让我更有勇气迎接未来的各种挑战，以后我会继续努力，成为更加优秀的人。