软件开发环境下API应用安全的实现与研究

Implementation and Research of API Application Security in Software Development Environment

黄　健（中国联通湖北分公司，湖北 武汉 430040）
Huang Jian（China Unicom Hubei Branch，Wuhan 430040，China）

摘　要：
API是数据交换的重要途径，其安全问题不容忽视。从基于API应用安全技术的身份验证、访问控制、消息加密及攻击检测等几个方面，深入探讨了在软件开发环境下API应用安全的实现方式与设计思路，以保障软件开发环境下API接口的使用安全，从而提升软件开发质量。

关键词：
软件开发；API应用安全；身份验证；访问控制；消息加密；攻击检测
doi：10.12045/j.issn.1007-3043.2024.08.008
文章编号：1007-3043（2024）08-0039-05
中图分类号：TP311.5
文献标识码：A
开放科学（资源服务）标识码（OSID）：

Abstract：
API is an important means of data exchange，and its security issues cannot be ignored. It explores in depth the implementation and design ideas of API application security in software development environments from several aspects such as identity authentication，access control，message encryption，and attack detection based on API application security technology，in order to ensure the secure use of API interfaces in software development environments and improve software development quality.

Keywords：
Software development；API application security；Identity authentication；Access control；Message encryption；Attack detection

引用格式：黄健. 软件开发环境下API应用安全的实现与研究[J]. 邮电设计技术，2024（8）：39-43.

1 概述

在数字化时代，软件开发需求增大。在软件开发过程中，需要考虑数据传输与交互的安全问题，应用程序编程接口（Application Programming Interface，API）是数据传输与交互的主要方式[1]。不安全的API接口环境可能带来敏感数据泄露、未经授权的访问和拒绝服务攻击等风险。因此，API安全技术对保障用户数据安全和API接口正常运行至关重要[2-3]。探索基于API安全技术的实现方法，对于提高软件产品交付质量具有重要意义。

收稿日期：2024-07-02

2 方案介绍

纵深防御是一种常用综合性的安全策略，这是在网络安全法中等级保护2.0实施指南里提到的。API安全设计也采用纵深防御策略，通过身份验证、访问控制、消息加密、攻击检测共同构成了API应用多层次安全防护措施。以身份验证为基础，确保只有合法的用户或系统能够访问API；在此基础上进一步细化权限管理，通过访问控制确保经过验证的实体只能访问和操作被允许的资源；通过消息加密保护数据在传输过程中的安全，防止被窃听和篡改；最后结合攻击检

测技术的实时监控和响应能力，确保能够及时发现和阻止潜在的安全威胁。通过这些技术的综合应用，可以有效保护API的安全，给软件开发者提供一个安全可靠的开发环境。

2.1 身份验证

在软件开发过程中认证机制是保障API应用安全的首要方式，其目的是确定API请求的来源和合法性。

2.1.1 实现方式

API应用安全的身份验证技术有很多种，主要包括基于令牌的身份验证、用户名密码身份验证和JWT（JSON Web Token）身份验证等。

a）基于令牌的身份验证：这种身份验证方式需要客户端和服务器之间传输安全凭证，客户端向服务器发送用户名和密码，服务器验证通过后会发放一个令牌。客户端在发送请求时会携带令牌，服务器根据令牌验证客户端的身份，令牌可以是JWT或其他的令牌。这种身份验证技术的优点是安全性高，缺点是实现较复杂，需要在客户端和服务器之间传递安全凭证，因此消耗更多的资源[4-5]。同时，安全性会受令牌失效期限的影响，。

b）用户名密码的身份验证：验证系统会通过用户名和密码，来验证请求者的身份。这是一种最传统的验证方式，适合于小型项目或者对于安全性要求不高的项目，其优点是简单易用，实现成本低，用户只需要输入他们的用户名和密码，就可以访问他们有权访问的资源。其缺点是用户名和密码需要存储在客户端，容易被泄露[6]。

c）JWT是一种基于JSON的轻量级身份验证令牌，它可以在各方之间安全地传输信息。它可以用来存储一些额外的信息，例如用户名、用户ID、权限等，以确保它们的安全性[7-8]。JWT可以让服务器验证客户端发送的请求是否是合法的，以防止跨站请求伪造攻击，这样可以有效地防止攻击者伪造请求，访问用户的数据[9-10]。JWT在HTTP头中传输，而无需在实际请求中发送大量数据。它还可以跨域进行认证，从而节省时间和网络带宽[11]。

2.1.2 实现流程

身份验证流程如图1所示。

通过身份验证对用户请求进行校验，并将用户请求分为登录注册请求和其他请求2类。若判断请求路径为登录或注册请求，则生成JWT令牌，并将令牌传入header中，便于之后用户再次发送请求时校验。若

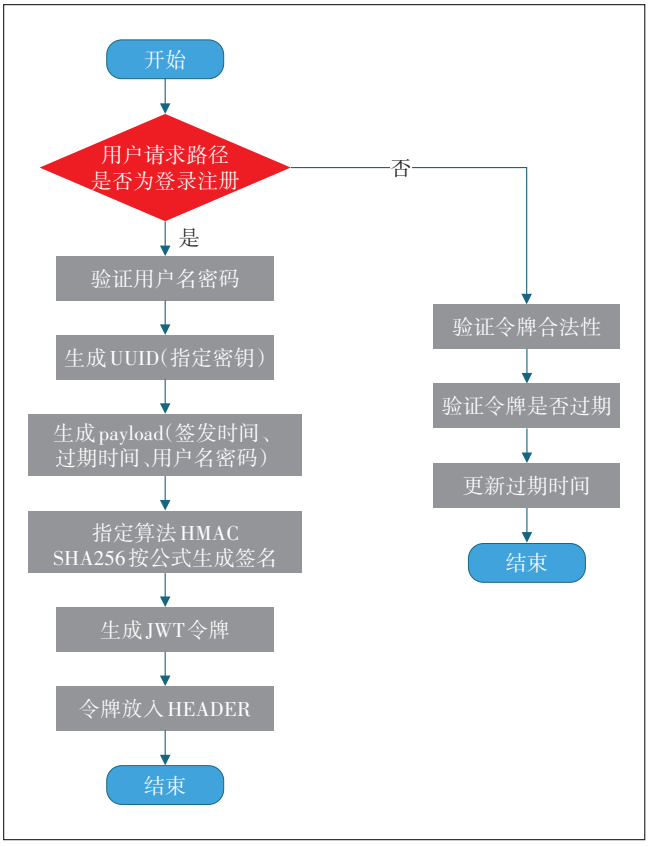

图1　身份验证流程

是判断请求路径为其他请求，则开始验证令牌合法性和过期时间，判断成功后再更新令牌过期时间，继续处理用户请求[12-13]。

API安全身份验证确保了软件开发环境下只有授权用户能够访问API资源，从而使API免受未经授权的访问和滥用。通过使用不同的身份验证方式，可以增强API的安全性，防止恶意攻击和数据泄露，并保护用户的隐私和机密信息。因此，在设计和部署API时，应该充分考虑身份验证的重要性，并采取相应的措施来确保API的安全性和可靠性。

2.2 访问控制

在软件开发过程中，用户角色的权限控制是需要考虑的一个重要安全问题。访问控制机制是限制API访问权限的一种方式，它可以确保API的数据仅被授权的用户或应用程序访问。

2.2.1 实现方式

常见的访问控制有基于访客的身份认证、基于资源的访问控制和基于属性的访问控制。

a）基于访客的身份认证（VisitorIdent Ificationand Authentication，VIA）。VIA是一种以访客身份认证为

基础的访问控制方式，其特点是可以细化到每一个用户的访问权限，即可以精准控制每一个访客的访问权限。

b）基于资源的访问控制将访问权限与资源进行关联。根据用户是否具有访问某个资源的权限来确定用户是否可以访问该资源。因此它可以更加精确地控制用户的访问权限，用户只能访问具有访问权限的资源，这样可以避免用户误访问或滥用系统资源的问题[14-15]。

c）基于属性的访问控制将用户的访问权限与用户的属性联系起来，根据用户的特征来决定用户是否有资格访问某个资源，以此来限制用户对资源的访问。因此它可以更加精细地控制用户的访问权限，根据不同的用户属性来设置不同的访问策略，而且能够更加灵活地管理用户的权限。同时，基于属性的访问控制还可以实现跨系统的权限管理，这使其成为一种非常实用的访问控制方式。

2.2.2　实现流程

访问控制的流程如图2所示。

访问控制是保障软件开发环境下API应用安全的重要手段，其中，基于访客身份认证可以细化到每个用户的访问权限；基于资源的访问控制可以精确地控制用户的访问权限；基于属性的访问控制可以实现跨系统的权限管理。这些访问控制技术增强了软件应用的安全性和可靠性。

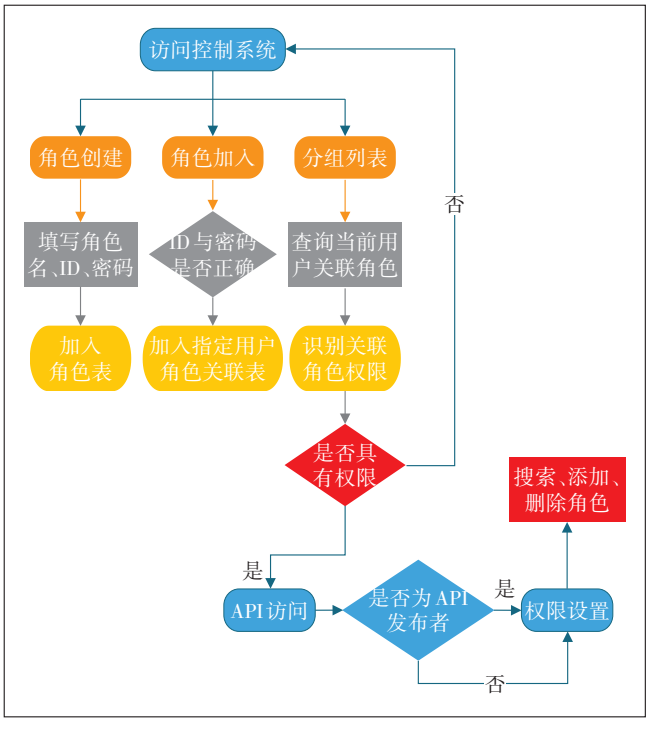

图2　访问控制流程

2.3　消息加密

加密机制是保护API数据传输安全的一种方式，它可以保护API数据在传输中不被窃取或篡改。在数据传输过程中，常常需要使用加密机制确保数据的机密性和完整性。

2.3.1　实现方式

消息加密应用于API的消息传输期间。当客户端在前端发送请求时，设置拦截器拦截请求并按加密方法的流程对请求中的数据加密，然后转换为Base64编码继续处理请求消息。从服务端发回的响应也使用拦截器拦截并对响应体中的数据进行解密，然后将Base64的编码格式转换为字符串，继续处理响应。

首先需要创建一个Crypto-js对象。Crypto-js是一个JavaScript库，使用者需要引入Crypto-js的文件，并选择加密算法。为了保证加密信息的安全性，可以通过Crypto-js对象的setKey()函数设置加密密钥。

在前端需要将待加密字符串转换为二进制，作为加密密钥。Crypto-js提供了encrypt()函数，用于加密数据。该函数的参数包括加密字符串和密钥。加密过程中，使用KeyGenerator和Cipher实现加密，并将加密后的byte数组进行Base64编码转换为字符串。同时，加密的参数word转换为二进制字符串，并使用之前设置的密钥进行加密。加密模式采用ECB，并进行PKCS5Padding数据填充。加密完成后，使用相同的密钥对加密后的字符串进行解密，并将解密后的字符串转换为utf-8编码，最后返回解密后的字符串。

在后端建立一个Rewrite Function接口，实现应用重写的功能。接收到的加密数据body首先调用AesEncryptUtils.encrypt()方法进行加密，然后将加密后的数据以Mono的方式返回。在Rewrite Function接口的apply方法中，接收前端传入的参数（body），并使用AesEncryptUtils.decrypt()方法对其进行解密处理，最后返回一个Mono<String>对象，其中包含解密后的数据。

最后，将解密后的数据发送至接收方。接收方收到数据后，可以使用Crypto-js解密密文，从而获取原始数据。

2.3.2　实现流程

消息加密的流程如图3所示。

Crypto-js是一个强大的Java Script加密算法库。

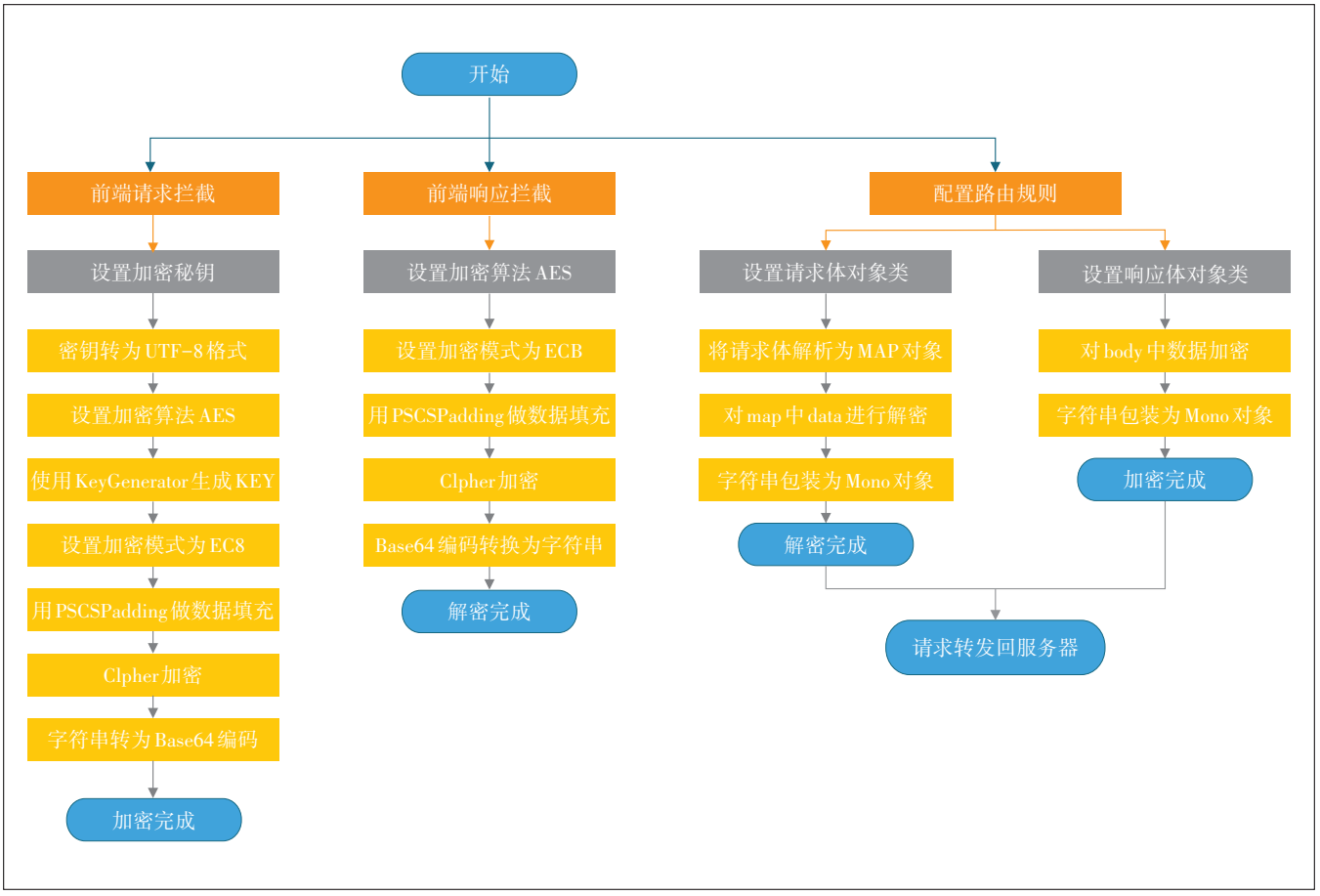

图3　消息加密流程

它可以支持多种密码算法，用于API应用中的消息加密，可以确保数据传输的安全性和完整性。这种技术可以有效地防止数据被篡改，从而确保用户的信息安全。

2.4 攻击检测

在软件开发环境下，为了验证软件运行环境的安全性，常常对软件API应用接口做周期性的攻击检测。

2.4.1 实现方式

为了达到更准确、更高效、更智能的测试效果，API应用接口的攻击检测通常会与2种深度学习模型相结合。这2种学习模型分别是长短时记忆（LSTM）和卷积神经网络（CNN），LSTM是一种特殊的循环神经网络，它可以在时间序列数据上进行长期依赖性的学习。它可以通过对大量的历史时间序列数据进行分析，来学习攻击行为的特征，从而检测出攻击行为。同时，它可以捕捉网络安全数据中的长期依赖性和短期依赖性，从而更好地检测攻击行为。LSTM模型由一系列记忆细胞组成，这些记忆单元可以提取网络安全数据中的特征，并学习网络安全数据中的规则，从而准确地预测攻击行为。

CNN是一种特殊的深度学习结构，它可以从输入图像中提取特征，并用于识别和分类任务。在网络攻击检测方面，CNN可以被用来识别网络攻击行为，并给出相应的预测结果。CNN的检测原理是首先将网络流量的数据转换成图像，然后将其作为CNN的输入层，图像经过卷积层、池化层和全连接层处理，从而提取出网络流量中的特征，最后将特征作为输出层的输入，并使用多层神经网络进行训练，以输出是否存在网络攻击的预测结果。CNN可以用于计算各种复杂的函数，并可以学习复杂的模式，可以从多个角度分析网络流量，从而提高检测的准确性。

2.4.2 实现流程

训练流程如图4所示。

3 总结

API应用安全是软件开发过程中的重要组成部

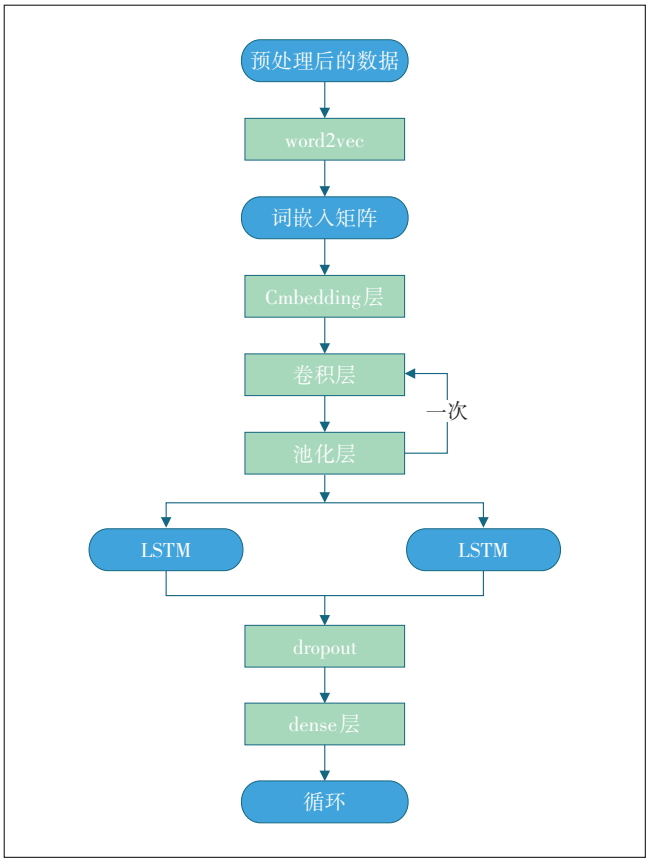

图4　训练流程

分。本文从多个角度探讨了API应用安全技术，包括基于身份验证、访问控制、消息加密及攻击检测等方面，并提供出了新的实现方式与设计理念。具体包括令牌、授权码、用户名密码、JWT的身份验证技术，多类型访问控制技术，使用Crypto-js加解密技术、基于深度学习模型的攻击检测技术，这些研究成果对于提升软件开发质量具有重要意义。展望未来，我们应该继续关注API应用安全问题，不断优化和完善相关技术，以确保软件应用中API接口的安全性和稳定性。

参考文献：

[1] ZHANG P, PEI Y N. A technology of user access-control table and identity authentication based on USB in LAN[C]//2010 International Conference on Biomedical Engineering and Computer Science. Piscataway: IEEE, 2010: 1-3.

[2] HONG C H, HEO J, JANG J G, et al. Quantum identity authentication with single photon[J]. Quantum Information Processing, 2017, 16(10): 236.

[3] HU C T, FERRAIOLO D F, KUHN D R, et al. Guide to attribute based access control (ABAC) definition and considerations: 800-162 [R/OL]. [2024-01-30]. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=914795.

[4] FUCHIDA S, YAMAGUCHI S, TAKAKURA H. Secure Apl: a framework for securing enterprise API gateway[C]//IEEE 4th Internatioal Conferenceon FutureInternet of Thingsand Cloud. 2016: 182-192.

[5] JIA J D, QIU X F, CHENG C. Access control method for Web of things based on role and SNS[C]//2012 IEEE 12th International Conference on Computer and Information Technology. Piscataway: IEEE, 2012: 316-321.

[6] CHEN H C. A cooperative RBAC-based IoTs server with trust evaluation mechanism[C]//IoT as a Service. Cham: Springer, 2018: 36-42.

[7] NING Y E, ZHU Y, WANG R C, et al. An eficient authentication and access control scheme for perception layer of Internet of Things[J]. Applied Mathematics & Information Sciences, 2014, 8(4): 1617-1624.

[8] ZHANG G P, GONG W T. The research of access control based on UCON in the Internet of Things[J]. Journal of Software, 2011, 6(4): 724-731.

[9] OUECHTATI H, AZZOUNA N B. Trust-ABAC towards an access control system for the Internet of things[C]//Green, Pervasive, and Cloud Computing. Cham: Springer, 2017: 75-89.

[10] 林智强，吴浩，张明，等. 基于拓扑的层次结构消息加密技术[J]. 计算机工程，2019，45(2)：77-80.

[11] BECKMAN M, MICHAEL F C, GILLESPIE W M. Elliptic curve cryptography based message encryption[J]. IEEE Transactions on Information Theory, 2015, 61(6): 3238-3254.

[12] BROCK B, WILFORD E L. Merkle tree protocol based API gateway MessageIntegrity verification[J]. IEEE Transactions on Information Forensics&Security, 2018, 13(11): 2842-2854.

[13] ERIK M L. Public key cryptography and message authentication[EB/OL]. [2024-01-30]. http://www. parkjonghyuk. net/lecture/2016-1st-lecture/networksecurity/erik.pdf.

[14] SADEGHIAN A, ZAMANI M, MANAF A A. A taxonomy of SQL injection detection and prevention techniques[C]//2013 International Conference on Informatics and Creative Multimedia. Piscataway: IEEE, 2013: 53-56.

[15] WARRENDER C, FORREST S, PEARLMUTTER B. Detecting intrusions using system calls: alternative data models[C]//Proceedings of the 1999 IEEE Symposium on Security and Privacy (Cat. No.99CB36344). Piscataway: IEEE, 1999: 133-145.

作者简介：
黄健，高级工程师，主要从事网络信息安全的规划、建设与运营工作。