基于 SM2 和 OAuth2.0 的强安全身份认证方案

陈艺琳,左黎明,郝　恬,罗娇燕
(华东交通大学 理学院,江西 南昌 330013)

摘　要:在网络技术广泛应用的时代,网络安全问题的重要性越来越显著,同时网络安全问题也越来越突出。身份认证技术是确保网络安全的重要手段。在 API(Application Programming Interface)控制访问中,OAuth2.0 协议兼具用户资源授权和委托访问控制方法,在国内外各大互联网厂商中应用较广。但是由于开发者未能严格遵守 OAuth2.0 协议的规范,导致数据来源不可靠性问题层出不穷。国密 SM2 数字签名算法是国内自主研发的基于 ECC 国际标准的改进算法,改进了明文编码问题并且具有更高的计算效率。为解决 API 身份认证技术中存在的安全问题,提出一种基于 OAuth2.0 的强安全身份认证方案,在其协议设计中使用了国密 SM2 数字签名,在无需密码的情况下完成第三方授权及认证,实现对受保护资源的授权和控制访问。结果表明,方案在保留了传统身份认证方案所具有的安全性的同时,兼具了抗重放攻击、防中间人攻击和抗伪造性的特点,且数据交互效率没有明显降低。

关键词:OAuth2.0 协议;SM2;数字签名;身份认证;授权
中图分类号:TP309.7　　**文献标识码**:A　　**文章编号**:1673-629X(2023)07-0126-06
doi:10.3969/j.issn.1673-629X.2023.07.019

Strong Security Authentication Scheme Based on SM2 and OAuth2.0

CHEN Yi-lin,ZUO Li-ming,HAO Tian,LUO Jiao-yan
(School of Science,East China Jiaotong University,Nanchang 330013,China)

Abstract:In the era of widespread network technology,the importance of network security issues is becoming more and more significant,while network security issues are also becoming more and more prominent. Authentication technology is an important means to ensure network security. In the API (Application Programming Interface) control access,OAuth2.0 protocol has both user resource authorization and delegated access control methods,which are widely used in major Internet vendors at home and abroad. However,due to the failure of developers to strictly comply with the specifications of OAuth2.0 protocol,the problem of unreliability of data sources has emerged one after another. The SM2 digital signature algorithm is an improved algorithm based on the ECC international standard developed by China,which improves the plaintext encoding problem and has higher computational efficiency. In order to solve the security problems in API authentication technology,we propose a strong security authentication scheme based on OAuth2.0,which uses SM2 digital signature in its protocol design to complete third-party authorization and authentication without passwords,and realize the authorization and control access to protected resources. The results show that the scheme retains the security of traditional authentication schemes while combining resistance to replay attacks,man-in-the-middle attacks,and forgery resistance,with no significant reduction in data interaction efficiency.

Key words:OAuth2.0 protocol;SM2;digital signature;authentication;authorization

0　引　言

随着互联网技术的发展,各类应用程序对用户身份认证的要求也越来越高。用户身份认证技术是网络信息安全体系中至关重要的一环,“以认证为核心”的零信任架构成为 SaaS 等网络服务与物联网安全[1]的必要技术。使用 token 作为身份认证技术在 API 接口中最为常见,基于此出现了一系列网络通信协议,其中 OAuth2.0 协议因其协议简单清晰而被广泛应用。然而,由于开发者在部署协议的过程中为了最小代价整合资源,并没有严格遵守协议规定,这就造成了一系列针对 OAuth2.0 协议的攻击。如淘网址认证登录漏洞(乌云编号 wooyun-2012-011104)[2]、由于 OAuth 2.0 无绑定 token 问题导致某 APP 可任意进入他人账号[3](乌云编号:wooyun-2013-017306)、2021 年最新微软

收稿日期:2022-08-20　　**修回日期**:2022-12-22
基金项目:江西省教育科技项目(GJJ200626,GJJ210625)
作者简介:陈艺琳(2000-),女,硕士研究生,研究方向为信息安全;通讯作者:左黎明(1981-),男,副教授,硕士,硕导,CCF 会员(E20-0013632M),研究方向为信息安全、非线性系统。

和 GitHub OAuth 实施漏洞导致重定向攻击[4]等。不法分子通过身份认证漏洞窃取公民隐私,威胁公民个人信息安全。随着开发者安全意识的提高,客户端应用程序对数据传输协议进行了更加严格的规定。但是在传统的用户名密码登录模式中,一旦用户的用户名和密码遭到泄露,攻击者仍然可以伪造用户身份进行数据访问。即使没有密码,攻击者仍然可以通过密码爆破、字典撞库等方式窃取用户身份,侵犯公民数据隐私[5]。

针对 OAuth2.0 协议产生的安全问题,国内外学者曾进行过多方面的研究。2014 年,S. Pai 和 Y. Sharma[6]等人针对应用了 OAuth2.0 协议的 Alloy 框架进行了分析,并验证其中存在的安全漏洞。CJ Mitchell[7]分析研究了中国基于 SSO(Single Sign On)单点登录的 OAuth2.0 协议,总结出可能存在 CSRF 漏洞和用户身份绑定缺陷逻辑漏洞。2019 年,刘奇旭[8]研究了面向 OAuth2.0 授权服务 API 的账号劫持攻击威胁检测,设计并实现了面向 OAuth2.0 授权服务 API 的账号劫持攻击威胁检测框架 OScan。2021 年,陈佳[9]研究了电力系统下微服务架构的安全性,设计了一种基于 JWT 规范的令牌认证方案,授权机制基于 OAuth2.0 授权协议扩展实现。

针对 OAuth2.0 协议的一些安全问题,该文提出了一种基于国密 SM2 和 OAuth2.0 强身份认证方案,将认证参数通过数字签名进行保护,实现了对用户身份的安全认证与安全控制访问。有效保证了数据完整性、数据机密性、不可抵赖性、消息新鲜性与来源可靠性。

1　OAuth2.0 协议及其攻击

1.1　OAuth2.0 协议

OAuth2.0 是一个开放的标准,通常拥有四个角色参与认证流程:资源所有者、资源服务器、客户端、授权服务器[10]。其中,资源拥有者为给予受保护资源访问权限的主机或者用户;授权服务器为在对资源拥有者进行身份认证并获得授权后,为客户端提供访问令牌;客户端为通常指代理用户发起受保护资源请求的客户端应用程序;资源服务器为存储受保护资源,接受授权服务器提供的访问令牌的服务器。

在客户端、资源管理器、授权管理器、三方不互信的情况下,OAuth2.0 协议允许资源所有者通过授权服务器授权第三方应用程序访问存储在资源服务器上的受保护的信息,而不需要共享密码。常见的授权服务器有国外的 Facebook、Google,国内的腾讯、新浪等。OAuth2.0 一般情况下有四种模式,即:授权码模式、简化授权模式、密码模式、客户端凭证模式。OAuth2.0 协议中四个角色的交互流程如图 1 所示。

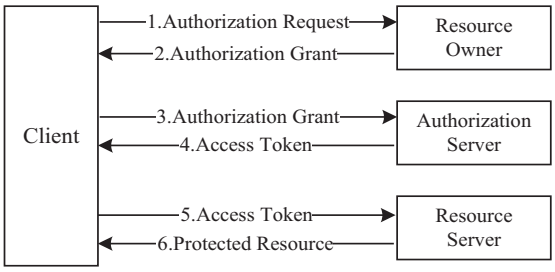

图 1　OAuth2.0 协议交互流程

其中,授权码模式一般用于带有 Server 端的应用程序。其认证流程如下:

(1)客户端通过用户代理去授权服务器进行授权,携带客户端标识及重定向 URI,response_type 值为 code;

(2)资源拥有者进行授权;

(3)接收到资源拥有者的确认授权后,授权服务器携带授权码请求客户端的重定向 URI,客户端提取授权码,再次向授权服务器请求 access_token;

(4)授权服务器返回给客户端 access_token;

(5)客户端携带 access_token 访问资源服务器;

(6)资源服务器向授权服务器验证 access_token,并向用户代理响应请求。

1.2　针对基于 OAuth2.0 认证方案的攻击

OAuth2.0 协议在部署时,为了让平台方以最小的代价整合所有资源,协议对许多方面并不做强制要求,这就导致一些平台的开发者在开发过程中由于安全意识薄弱,并没有正确地应用 OAuth2.0 协议,造成了许多安全问题[11]。OAuth2.0 协议中,各个角色所承担的功能不同,于是可以划分出以下几种攻击方式。该文就其中一些典型漏洞做出解释。

(1)针对客户端的攻击。

针对客户端的 CSRF(Cross - Site Request Forgery)[12]攻击流程如图 2 所示。攻击者伪造合法请求,诱导完成了身份授权的受害用户触发该请求,从而达到窃取令牌的目的。漏洞的原因是客户端应用程序错误使用或未使用 state 参数,该参数值是一个随机字符串,用于维持请求和回调之间的状态。攻击完成的前提是受害者在客户端应用程序完成授权。由于请求都来自客户端应用程序而非资源拥有者,而且授权码是资源拥有者与客户端的唯一标识,授权服务器并不知道授权码来自哪个资源拥有者。

(2)针对授权服务器的攻击。

redirect_uri 被篡改为恶意链接重定向地址 redirect_uri,是指授权服务器将授权通过后的响应发送到的重定向 URI。若客户端在注册时使用了不严格的重定向地址,或者授权服务器未对重定向地址进行严格校验,

就会导致重定向地址被篡改为另外的回调 URL,甚至在对重定向地址进行了校验的情况下页可以被绕过。利用这种 URL 跳转或 XSS 漏洞,可以获取到相关授权 token,危害到目标用户的账号权限。下面进行详细说明。

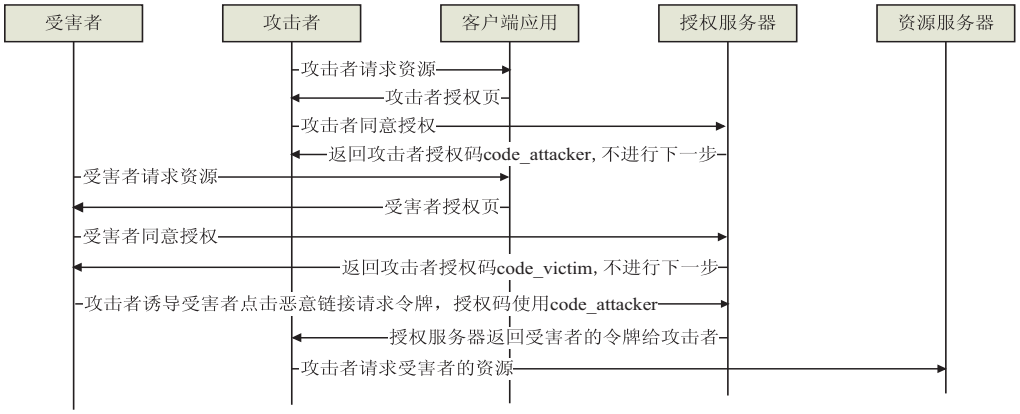

图 2 针对 OAuth2.0 协议客户端的 CSRF 攻击流程

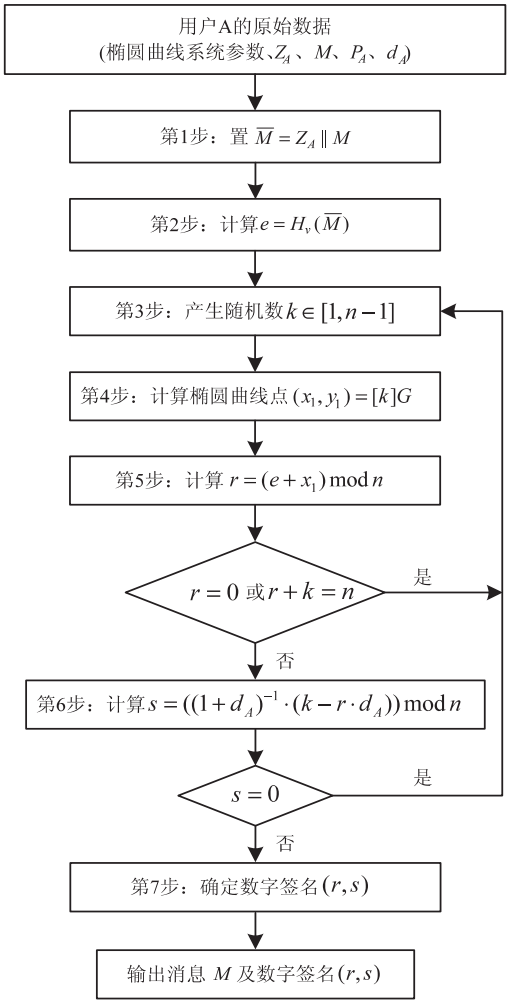

图 3 SM2 数字签名算法流程

若客户端在注册时使用了不严格的重定向地址,就可能造成授权服务器在校验时因校验不完整的而绕过。若授权服务器只对重定向地址的根域做了匹配,即只要请求的 URI 中包含注册的 URI 就认为是合法请求,那么授权码和 state 就会因重定向地址篡改而被发送至攻击者构造的恶意地址;获取到授权码之后,攻击者便可以使用受害者的授权码和 state 伪造身份向客户端发起正常回调请求,客户端校验后向授权服务器发送令牌申请,完成后攻击者便可访问受害者的资源。

1.3 国密 SM2 数字签名

国密 SM2 算法是国家密码局于 2010 年颁布的椭圆曲线公钥密码算法,在政府、银行和关键基础措施系统中得到广泛应用,目前国家对国密算法使用已经有强制性要求[13]。使用 SM2 算法进行数字签名主要有三个步骤:密钥生成、签名生成和验证签名。国内已有众多将 SM2 签名算法或其改进算法应用于各类认证方案中的实例[14]。

参数说明和密钥生成见文献[15],签名生成和验证过程如下:

(1)签名生成。

国密 SM2 数字签名生成算法流程如图 3 所示。用户 A 具有长度为 $\mathrm{entlen_A}$ 比特的可辨别标识 $\mathrm{ID_A}$,记 $\mathrm{ENTL_A}$ 是由整数 $\mathrm{entlen_A}$ 转换而成的两个字节,杂凑值 $Z_A=H_{256}(\mathrm{ENTL_A}\ \|\ \mathrm{ID_A}\ \|\ a\ \|\ b\ \|\ x_G\ \|\ y_G\ \|\ x_A\ \|\ y_A)$。

(2)签名验证。

国密 SM2 数字签名算法的验证流程如图 4 所示。设接收到的消息 M' 及其数字签名 (r',s'),接收用户用图 4 所示流程验证数字签名的合法性。

2 强安全方案设计及其关键实现

2.1 强安全方案设计

身份认证一般有三种场景[16],基于桌面应用的客户端模式(C/S)[17]、基于移动端 app(APP/S)客户端模式[18]和基于浏览器的客户端模式(B/S)[19]。三种场景下均采用国密 SM2 算法进行数字签名。三端交互的具体流程如图 5 所示,其中 C 为客户端,S_1 为授

权服务器, S_2 为资源服务器。

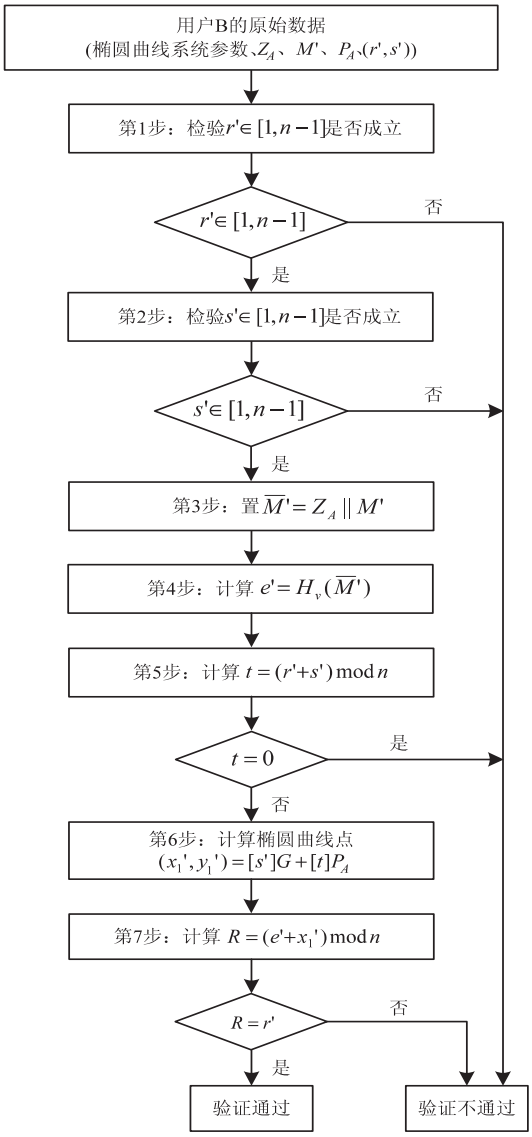

图 4　SM2 签名验证算法流程

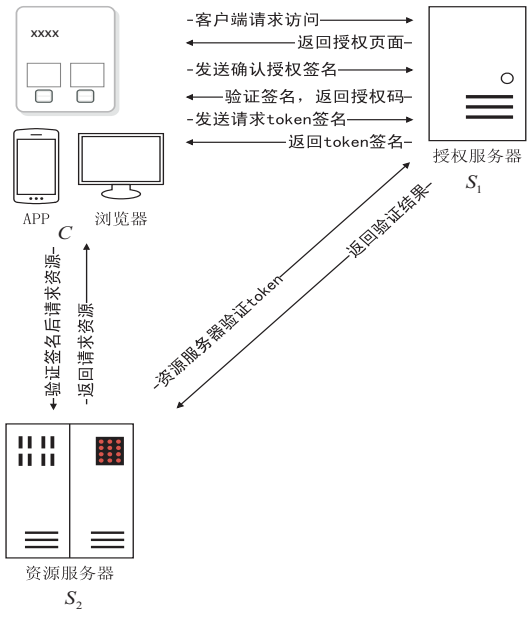

图 5　C/S、B/S、APP/S 下的身份认证流程

(1) $C \rightarrow S_1$。

首先客户端发送访问申请,若未授权,则授权服务器返回授权页面。

(2) $C \rightarrow S_1$。

用户确认授权,签名内容

sig_1 = uid ‖ client_id ‖ redirect_uri ‖ datestate ‖ response_type 。

(3) $S_1 \rightarrow C$ 。

授权服务器验证签名,并返回授权码信息签名,签名内容

sig_2 = code ‖ uid ‖ client_id ‖ state ‖ date ‖ redirect_uri 。

(4) $C \rightarrow S_2$。

客户端请求 token ,签名内容

sig_3 = code ‖ uid ‖ client_id ‖ state ‖ date ‖ grant_type ‖ redirect_uri 。

(5) $S_1 \rightarrow C$ 。

授权服务器验证签名,并返回 access_token ,签名内容

sig_4 = state ‖ access_token ‖ uid ‖ client_id ‖ date ‖ token_type 。

(6) $C \rightarrow S_2$。

客户端使用 access_token 请求资源服务器。

(7) $S_2 \rightarrow S_1$。

资源服务器将 access_token 发送至授权服务器进行验证。

(8) $S_1 \rightarrow S_2$。

授权服务器返回验证结果。

(9) $S_2 \rightarrow C$ 。

若验证成功,则资源服务器返回客户端请求的受保护资源。

2.2　授权码模式及其实现

授权服务器端统一采用 C#开发的 ashx 一般处理程序进行数据验证;带有 Server 端的客户端也采用一般处理程序 ashx 作为 redirect_uri 客户端重定向地址。在基于桌面应用客户端 C/S 架构下,使用 C#开发 WinForm 程序,并进行 SM2 算法的签名和验证。

```
C#客户端对消息进行签名:
TanSM2 sm2=new TanSM2();
String msg=uid+"#"+time+"#"+state+"#"+
redirect_uri+"#"+client_id+"#"+response_type;
sm2.TanSM2Sign(msg, user_prikey,
out sigtoCA);
服务端进行签名验证:
TanSM2 sm2=new TanSM2();
String msg=uid+"#"+time+"#"+state+"#"+
redirect_uri+"#"+client_id+"#"+response_type;
```

```
bool flag = sm2.TanSM2Verify
(msg,pubkey,sig);
```

第一步,客户端向授权服务器端发送请求签名,数据格式如图 6 所示。

```
POST /SM2verify.ashx HTTP/1.1
Content-Type: application/json;charset=UTF-8
Host: localhost:4170
Content-Length: 288
Expect: 100-continue
Connection: Keep-Alive

HTTP/1.1 100 Continue

{
  "uid": 1,
  "client_id": 12345,
  "redirect_uri": "http://localhost:4170/redirect.ashx",
  "response_type": "code",
  "time": "2021/12/17 16:16:18",
  "state": "43CEBA",
  "sigtoCA": "TorWdoz4b/
zzVlTucCTJEFKNeMeaFXZQ3TEa5t2vIlg=,GvLfKqbNqDYMj9mtWpzt0DUad3s1iijWWi6zctOThb4="
```

图 6 客户端向授权服务器发送请求数据格式

生成 5 位随机字符串 state,并加入时间戳 date。

第二步,授权服务器向客户端重定向地址发送授权码,数据格式如图 7 所示。服务端获取到 json 数据后,从中解析出加密后的密文参数,并使用服务端私钥进行解密。解密完成后,从数据库中查找该用户的公钥,并使用用户公钥对客户端发送的明文消息进行签名验证。验证成功后,授权服务器向重定向地址发送授权码 code。

```
POST /redirect.ashx HTTP/1.1
Content-Type: application/json;charset=UTF-8
Host: localhost:4170
Content-Length: 295
Expect: 100-continue
Connection: Keep-Alive

HTTP/1.1 100 Continue

{
  "uid": "1",
  "client_id": "12345",
  "code": "testcode",
  "response_type": "code",
  "time": "2021/12/17 16:20:12",
  "state": "YB54M1",
  "flag": "True",
  "Status": 1,
  "server_sig":
"WZCc9Gv8uAZd8279HzKnPT0H5BXPXC3bTvya5wtNzhQ=,ANHmmUzbOJsbvLDV/6eZCyuV+n/
Uz5YyczbxYU3BiLHe"
}HTTP/1.1 200 OK
```

图 7 授权服务器返回授权码数据格式

第三步,客户端携带授权码向授权服务器请求 access_token,请求数据格式如图 8 所示。重定向地址接收到授权码之后,解密并验证签名。验证通过后携带授权码向授权服务器/token 目录请求 access_token。

```
POST /token.ashx HTTP/1.1
Content-Type: application/json;charset=UTF-8
Host: localhost:4170
Content-Length: 316
Expect: 100-continue

HTTP/1.1 100 Continue

{
  "uid": "1",
  "client_id": "12345",
  "code": "testcode",
  "grant_type": "authorization_type",
  "time": "2021/12/17 16:20:12",
  "state": "YB54M1",
  "flag_code": "True",
  "Status": 1,
  "sigToCAForToken": "APs8ll8pMKS7RUCPYKf4hOnKNV4sJOKuO8ghMyz2SqpU,ANjg6vF/
jGKJ5B87aXDqJmsXGfoZWddXza5FShI29Fhe"
}HTTP/1.1 200 OK
```

图 8 客户端携带授权码申请令牌数据格式

第四步,服务端向客户端返回 access_token,返回数据格式如图 9 所示。服务端验证 code 后,向客户端返回 access_token。

```
{
  "uid" : "1",
  "client_id" : "12345",
  "access_token" : "access_token_test",
  "token_type" : "Bearer",
  "time" :"2021/12/17 16:20:12",
  "state" :"YB54M1",
  "flag_token" : "True",
  "Status" : 1,
  "sigTokenToClient" :
  "Ay92X8HYkA6GGJxDd0hlesrxfKgfwcIP4+PvTJgGDnO=,AJ4HeBndfBXbU7AE04ocr8
  979TkuAQdX80ULclPZc+l7",
  "client_flag" : "True"
}
```

图 9 授权服务器返回令牌数据格式

下一步客户端便可以使用 access_token 访问资源服务器上的受保护资源,资源服务器将请求转发至 CA 验证 access_token 的有效性之后便可将受保护资源返回给客户端。

3 安全性分析

3.1 基本分析

(1)数据机密性。

由于在数据传输过程中使用国密 SM2 签名算法对交互数据进行数字签名,并在授权服务器端进行签名验证,防止了攻击者恶意篡改数据。

(2)数据完整性。

授权服务器端会逐一验证签名中的参数和传递的参数是否一致,攻击者无法删除掉交互过程中的任意参数。

(3)不可抵赖性。

客户端会和用户都拥有自己的唯一私钥,并用客户端和用户的唯一私钥进行签名。授权服务器会使用私钥对应的公钥进行验证,保证了数据的不可抵赖性。

(4)消息新鲜性。

消息的新鲜性可有效抵抗重放攻击。在发送的消息中加入时间戳与随机字符串 state,授权服务器根据时间戳来确定消息的新鲜性。

3.2 对比分析

各个应用了 OAuth2.0 协议平台方案安全性对比分析如表 1 所示。

方案 1:微软和 GitHub OAuth;

方案 2:人人网 OAuth 2.0 授权;

方案 3:百度开放平台 OAuth 授权接口;

方案 4:基于国密 SM2 的 OAuth2.0 认证。

表 1 安全性对比分析

	方案 1	方案 2	方案 3	方案 4
数据完整性	√	√	√	√
数据机密性				√
不可抵赖性	√	√	√	√
消息新鲜性	√			√
来源可靠性	√	√	√	√

从表 1 中可以看出,方案 1、2、3 由于存在重定向 uri 篡改问题,在数据机密性和来源可靠性方面存在安全问题。方案 4 采用国密 SM2 算法对信息进行数字签名,有效保证了数据完整性、数据机密性、不可抵赖性、消息新鲜性与来源可靠性。

4　结束语

针对目前 OAuth2.0 协议中存在的数据来源性问题,提出了一种基于国密 SM2 的 OAuth2.0 身份认证方案。方案对可能会被篡改的信息如 uid、client_id、state、date 、redirect_uri 等进行数字签名,并在授权服务器端进行签名验证。随后针对 B/S、C/S、APP/S 三种场景下的身份认证进行仿真实验,实验表明该方案具有数据机密性、数据完整性、不可抵赖性、消息新鲜性,且传输效率未降低。

参考文献:

[1] 王志贺,骆　钊,谢吉华,等. 基于 SM2 密码体系的 SD 卡的电力移动终端安全接入方案[J]. 中国电力,2015,48(5):75-80.

[2] 乌　云. 淘网址 sina oauth 认证登录漏洞[EB/OL]. (2012-08-24)[2021-12-20]. http://www. wooyun. org/bugs/wooyun-2012-011104.

[3] 乌　云. 任意进入他人账号(OAuth 2.0 无绑定 token 问题)[EB/OL]. (2013-01-14)[2021-12-20]. http://www. wooyun. org/bugs/wooyun-2013-017306.

[4] Proofpoint. 微软、谷歌 OAuth 漏洞被用于钓鱼攻击[EB/OL]. (2021-12-09)[2021-12-20]. http://www. proofpoint. com/us/blog/cloud-security/microsoft-and-github-oauth-implementation-vulnerabilities-lead-redirection.

[5] 双　十. 黑客 Web 攻击的十大原因及十大抵御方法[J]. 网络与信息,2008,22(11):58-59.

[6] PAI S,SHARMA Y,KUMAR S,et al. Formal verification of OAuth 2.0 using alloy framework[C]//2011 international conference on communication systems and network technologies. Katra:IEEE,2011:655-659.

[7] LI W,MITCHELL C J. Security issues in OAuth 2.0 SSO implementations[C]//International conference on information security. Hong Kong:Springer,2014:529-541.

[8] 刘奇旭,邱凯丽,王乙文,等. 面向 OAuth2.0 授权服务 API 的账号劫持攻击威胁检测[J]. 通信学报,2019,40(6):40-50.

[9] 陈　佳. 一种基于 JWT 令牌认证的电力系统微服务认证授权方案[J]. 电工技术,2021(16):151-154.

[10] 钱君生,杨　明,韦　巍. API 安全技术与实战[M]. 北京:机械工业出版社,2021.

[11] 邱永哲. OAuth 2.0 授权协议常见安全问题及修复建议[J]. 无线互联科技,2018,15(7):45-47.

[12] 王丹磊,李长军,赵　磊,等. OAuth 2.0 协议在 Web 部署中的安全性分析与威胁防范[J]. 武汉大学学报:理学版,2016,62(5):411-417.

[13] 国家密码管理局. 国家密码管理局关于发布《SM2 密码算法使用规范》等 14 项密码行业标准公告[EB/OL]. (2012-11-22)[2021-12-20]. https://sca. gov. cn/sca/xxgk/2012-11/22/content_1002397. shtml.

[14] 陈荣征,邹　昆. 一种基于国产密码算法的身份认证方案[J]. 齐齐哈尔大学学报:自然科学版,2015,31(4):14-17.

[15] 国家密码管理局. 国家密码管理局关于发布《SM2 椭圆曲线公钥密码算法》公告附件 2[EB/OL]. (2010-12-17)[2021-12-20]. https://sca. gov. cn/sca/xxgk/2010-12/17/content_1002386. shtml.

[16] 左黎明,夏萍萍,陈祚松. 基于国密 SM2 数字签名的网络摄像头保护技术[J]. 信息网络安全,2018(5):32-40.

[17] 郭建伟. 实现 AnyConnect VPN 多重身份认证[J]. 网络安全和信息化,2021(9):138-144.

[18] 林国成,徐　策,王少平,等. 基于大数据的电网企业移动智能终端身份认证准入平台[J]. 自动化技术与应用,2021,40(12):146-148.

[19] 高保忠,杜首燕,李信治,等. 基于 OAuth2.0 协议的智慧校园认证系统研究[J]. 中国科学技术大学学报,2019,49(7):564-571.