Computer Science and Application 计算机科学与应用, 2021, 11(3), 579-587
Published Online March 2021 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2021.113059

Hans汉斯

基于Whisper的身份识别机制

吴鸿文，周　宇，杨振国，刘文印

广东工业大学计算机学院，广东 广州
Email: 12034408@qq.com, yuzhou1su@foxmail.com, yzg@gdut.edu.cn, liuwy@gdut.edu.cn

收稿日期：2021年2月20日；录用日期：2021年3月15日；发布日期：2021年3月22日

摘　要

用户一般需要使用帐户和密码来访问各种平台和系统，如果用户使用弱密码的话，虽然很容易记住，但比较容易受到攻击；如果使用强密码，则使用起来不太方便，不容易记住。所以，我们提出了一种基于区块链的去中心化身份认证机制。该机制使用以太坊的Whisper协议来取代http/https协议；更具体地的说，接入该机制的网站通过接收来自Whisper的内容来验证用户的身份信息，不再需要让用户填写用户名与密码进行验证。这种机制也能防御“重放攻击”、“网络钓鱼攻击”和“模拟攻击”，最后本机制与“OAuth2.0”，“OpenID”和“SAML”进行对比，在“网络钓鱼攻击”方面更比其他机制更优，能够很好的防御此类攻击。

关键词

区块链，以太坊，权威证明，密码，网络身份认证，单点登录

Blockchain with Whisper Protocol for Identity Authentication

Hongwen Wu, Yu Zhou, Zhenguo Yang, Wenyin Liu

School of Computers, Guangdong University of Technology, Guangzhou Guangdong
Email: 12034408@qq.com, yuzhou1su@foxmail.com, yzg@gdut.edu.cn, liuwy@gdut.edu.cn

Received: Feb. 20th, 2021; accepted: Mar. 15th, 2021; published: Mar. 22nd, 2021

Abstract

Users need to use accounts and passwords to access various platforms and systems. Weak passwords are easy to be remembered but vulnerable to attacks, while strong passwords are not easy-to-use. To this end, a blockchain-based and decentralized identity authentication mechanism without traditional passwords is proposed. Instead of using the http/https protocols, the Whisper

文章引用: 吴鸿文, 周宇, 杨振国, 刘文印. 基于 Whisper 的身份识别机制[J]. 计算机科学与应用, 2021, 11(3): 579-587.
DOI: 10.12677/csa.2021.113059

protocol in Ethereum is adopted. More specifically, the website verifies the identity information of the user by receiving a content of a Whisper envelope, thus the website does not need to provide a web interface, in order to verify the identity information of the user. The proposed identity authorization process in a decentralized manner has been verified to defend against replay attack, phishing attack and impersonation attack, compared with OAuth2.0, OpenID and SAML.

Keywords

Blockchain, Ethereum, Proof of Authority, Password, Web Identity Authentication, SSO

1. 引言

随着互联网的快速发展，也诞生了各种各样的服务，例如电子邮件、购物、社交聊天、银行服务，娱乐等。人们可能同时拥有相当多的网络身份(往往采用帐户和密码进行登录)，从而导致一个称为“密码疲劳”的安全问题。大多数应用程序都会要求用户在访问其资源之前进行身份验证，为了方便起见，人们通常使用比较容易记住的弱密码。但是，一个普通的 CPU 可以在 20 秒内破解包含 6 个小写字符的密码；对于更长的密码，黑客通常使用可预测的单词对其进行破解[1]。

单一登录机制(SingleSignOn, SSO)是一种广泛使用的机制，它通过依赖诸如 Facebook、QQ、微博等提供的授权接口对用户身份进行验证。SSO 减轻了密码疲劳，使访问应用程序更加轻松快捷，但是，SSO 是集中式身份验证机制，当提供者的服务出现宕机的话就无法完成识别与授权操作。这些问题促使我们寻找更加有效且更加安全的身份识别服务解决方案。

在本文中，我们提出了一种基于区块链的去中心化身份识别机制。更具体地说，每个网站都是区块链网络的一个节点，这样避免了单点故障。Whisper [2]是以太坊基于身份的消息传递协议，本机制中不同节点之间的通信就采用 Whisper 协议，并且内容都是已加密的，以确保安全性。IPFS [3]是一个具有持久性且去中心化存储和共享数据的网络系统；由于 Whisper 包的大小可能会影响其在区块链网络上广播的速度，节点先把需要传递的内容上传至 IPFS 服务，IPFS 则返回该内容的 hash 值，节点之间传递的内容仅为 hash 值，由于 Whisper 仅广播 hash，所以传播的内容就变得很小。使用 IPFS 还具备另外一个优点就是数据不会丢失，传统集中式服务的话，如果内容丢失则很难再找会，但是采用去中心化存储系统(例如 IPFS [3]，Storj [4]，Sia [5]等)不依赖中央服务提供商，使得内容存储更加安全。

本文的主要贡献概述如下：

1) 我们提出了一种去中心化身份识别机制，该机制不再依赖于中央节点并且可以做到高并发；使用此机制的网站不再需要提供一个登录接口(登录接口往往也是会比较容易受到攻击的)，这可以使网站更加安全。

2) 我们针对该机制做了相关的攻击实验，在实验结果中发现可以更好地抵御“重放攻击”、“网络钓鱼攻击”和“模拟攻击”等攻击方式。

2. 预备知识

在传统的基于密码身份识别与授权方案中，通常采用输入用户名和密码来验证用户的身份，尤其是在

Web 或 App 上。此方案容易实现且对用户方便，但是一旦用户名和密码被盗，用户的信息就会被盗，给用户带来重大的损失。

OAuth2.0 是用于身份识别与授权的行业标准协议，并允许服务在与其他服务交互时代表用户执行操作[6]。它使第三方应用程序可以代表资源所有者访问服务器资源，而无需共享其凭据(通常是用户名和密码)，这样对用户来说也是较安全的。OAuth2.0 专注于简化客户端开发人员的工作，同时为 Web 应用程序、桌面应用程序、移动端应用提供特定的授权流程。但是在 Avinash Sudhodanan 发表的这篇论文[7]中，提到在 133 个采用 OAuth2.0 机制的网站(选自 Alexa 全球排名最高的网站，属于三个不同等级范围)中，有 70%的网站未实施 CSRF 防御策略。网络钓鱼攻击[8] [9]是一种犯罪攻击，黑客模仿可信赖的实体(例如，知名网站)，并哄骗用户提供用户名、密码和/或其他个人信息；OAuth2.0 也不能很好的抵御网络钓鱼攻击[10]，黑客模仿搭建一个假冒“微信”、“QQ”、“微博”这样的网站，用户则很难分辨这是一个假冒的网站，有可能在假冒的网站输入用户名和密码，最后被黑客窃取了信息。

数字签名采用称为公钥基础结构(PKI)的标准，是具有比较高的安全性和通用性。数字签名可以在不安全的环境发送消息，并为该消息创建了一层验证和安全性；数字签名使接收者有理由相信该消息是由声明的发送者发送的。数字签名保证了消息或文档的内容在传输过程中没有被更改，接收者可以使用发送者的公钥验证签名的有效性，如果接收者事先知道发送者的公钥则不需要中央服务器获取发送者的公钥。

中心化身份认证是当前最常用的方法，其中中央节点具有最高的控制权，但是，一旦集中式服务器宕机，则身份识别机制就无法提供服务。区块链技术可以使身份识别机制去中心化，而不再依赖中心服务器。Blockstack ID [11]是一款去中心化的计算网络和应用生态系统，接入该机制的网站对用户身份识别时候则不再有中心化服务器宕机而无法完成身份识别的问题。uPort [12]是一个自我主权的身份和数据平台，用户能够向以太坊区块链注册一个全球唯一的标识符，从而使用户可以控制其身份、私钥、用户帐户和私有数据。目前 Blockstack ID 和 uPort 在去中心化的身份识别与授权机制都取得了一定的市场占有量。

3. 基于 Whisper 的身份识别机制

3.1. 架构设计

如前所述，基于密码的 Web 身份认证有被盗的风险，而类似 SSO 的中心化身份认证则依赖于中央节点的可靠性。为了验证不同节点通讯数据的有效性，我们需要一些权威节点进行验证，因此我们使用 POA 机制。本机制使用数字签名代替传统基于密码的方式。使用数字签名技术，用户登录不再需要输入密码，避免了由于密码在网络上传输时被偷取的风险。

我们提出的机制包含 4 个角色，即“权威节点”、“完整节点”、“轻型节点”、“IPFS 服务器”。权威节点负责数据上链并进行打包。在本机制中，智能合约的拥有者为权威节点，只有权威节点才能对智能合约进行修改的权力。如果网站信誉良好，则可以将其升级到权威节点，并且如果该网站的行为不当或存在作恶行为，那么则撤消其“权威节点”的职责。任何网站都可以成为完整节点，轻节点可以是移动应用程序，因此我们开发了一个“LoginAuth”移动端应用程序来存储用户的私钥。IPFS 服务器提供去中心化的高吞吐量分布式存储系统，IPFS 中存储的所有数据都经过加密，以确保数据安全。该机制中的所有角色都可以相互通信。更具体地说，“权威节点”，“完整节点”和“轻型节点”之间的通信采用以太坊的 Whisper 协议，而权威节点，全节点和轻型节点可以使用 https 访问 IPFS 服务，如图 1。

3.2. 基于区块链身份识别算法

本机制中的各个节点通信不是使用传统的 Http/Https，而是使用以太坊 Whisper 协议，传送的信息都是加密的，以广播的形式进行快速传播[13]。

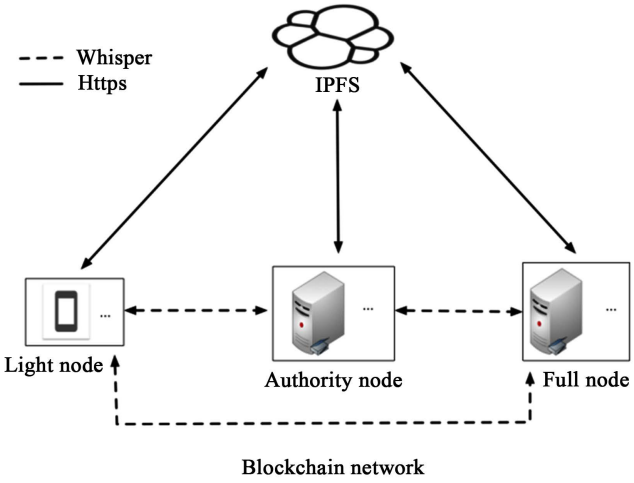

Figure 1. The architecture of the proposed mechanism
图 1. 架构概要

本机制创建“WebsiteManager”和“UserManager”这两个智能合约分别存储网站和用户的信息。WebsiteManager 合约包括域名和其对应公钥的信息。由于用户信息内容的大小相对较大，因此用户信息将通过其自己的公钥进行加密并上传到 IPFS，且 UserManager 合约仅存储来自 IPFS 的 hash，UserManager 合约存放用户的信息。只有“权威节点”有权限调用 WebsiteManager 合约和 UserManager 合约的修改方法修改数据。

例如，用户希望使用此机制登录网站。用户已将其信息上载到 IPFS，IPFS 返回对应密文的 hash，UserManager 合约保存了该 hash。WebsiteManager 合约存储网站的公共密钥。交互图如图 3 所示，它描述了用户身份识别过程。该机制的身份识别过程具有四个重要算法，分别是“生成 QR 码算法”，“QR 码签名验证算法”，“广播身份验证算法”和“验证身份算法”。

3.2.1. 生成 QR 码算法

QR 码主要用于将网站的登录信息传递给“LoginAuth”，网站的登录信息已二维码的形式呈现出来，“LoginAuth”通过扫描二维码，获得二维码的内容。在该算法中，我们将生成一个随机字符串作为 topic 参数，以使用以太坊 whisper 协议与网站进行通信。网站开始根据 topic 参数监听其对应频道的信息，然后生成当前时间戳以表示该登录时间；为了防止数据被篡改，网站使用私钥对数据进行签名，如图 2 (表 1 和算法 1)。

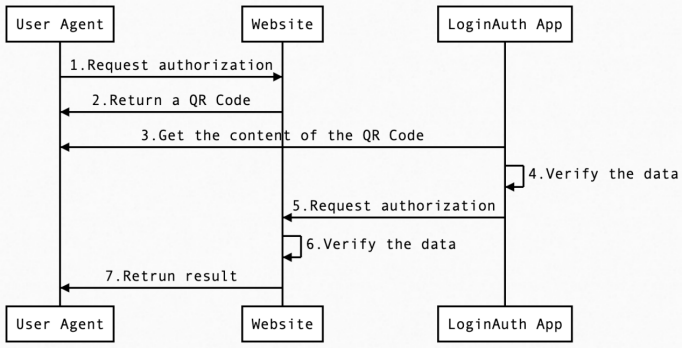

Figure 2. The interaction diagram of authenticate process
图 2. 身份识别的交互图

Table 1. Symbols and functions
表 1. 符号和方法描述

Symbol or function	Description
‖	与操作
M	网站域名
H()	hash 方法
SK_w	网站的私钥
PK_w	网站的公钥
SK_u	用户的私钥
PK_u	用户的公钥
Rand()	生产随机字符串的方法
Time()	生产当前时间戳的方法
Sign()	签名方法
Verify()	验证方法
Encrypt()	用公钥进行加密的方法
Decrypt()	用私钥进行解密的方法
UserManagerGet()	从 UserManager 合约中根据用户的公钥获取用户信息的方法
WebsiteManagerGet()	从 WebsiteManager 合约中根据网站域名获取网站信息的方法
IPFSPost()	上传信息到 IPFS 系统的方法
IPFSRequest()	从 IPFS 系统中获取信息的方法
WhisperBroadcast()	通过 Whisper 进行广播的方法

Algorithm 1. Generation QR code algorithm
算法 1. 生成 QR 码算法

```
1:  function GenQRCode()
2:      D1 = Rand()
3:      T1 = Time()
4:      D2 = H(D1‖T1)
5:      D3 = Sign(SKw, D2)
6:      return (D1, D3, T1, M)
```

3.2.2. QR 码签名验证算法

作为轻型节点的“LoginAuth”可以通过 WebsiteManager 合约获取网站的公钥，以验证内容的有效性。QR 码的有效期仅为几秒钟，可以保证 QR 码数据的可信度[13]，以下的算法设置的有效期为 10 秒，算法 2 中显示了其详细信息。

Algorithm 2. QR code signature verification algorithm
算法 2. QR 码签名验证算法

```
1:  function QRCodeVerify(D3, T1, M)
2:      T2 = time()
3:      if (T2 − T1 > 10) then
4:          return false
5:      PKw = WebsiteManagerGet(M)
6:      D4 = Verify(PKw, D3)
7:      return D4
```

3.2.3. 广播算法

采用 Whisper 发送的内容越小，那么再网络中传输就会越快，所以我们的算法旨把广播的内容体积最小化。“LoginAuth”使用该网站对应的公钥加密用户信息，并将其上传到 IPFS，“LoginAuth”则广播该 IPFS 返回的密文对应的 hash 即可。算法 3 中显示了详细信息。

Algorithm 3. Broadcast algorithm
算法 3. 广播算法

```
function Send(D1, D3, T1, M)
    if (QRCodeVerify() == false) then
        return false
    D5 = UserManagerGet(PKu)
    D6 = IPFSRequest(D5)
    PKw = WebsiteManagerGet(M)
    D7 = Encrypt(PKw, D6)
    D8 = IPFSPost(D7)
    T3 = time()
    D9 = H(D1||D8||T3)
    D10 = Sign(SKu, D9)
    M1 = {D8, D10, PKu, T3}
    WhisperBroadcast(D1, M1)
```

3.2.4. 身份识别算法

网站(“权威节点”或“完整节点”)通过接收 Whisper 协议收到内容来验证用户身份。此时，Whisper 的 topic 为随机字符串，与 QR 码算法 1 生成的随机字符串一致。如果身份验证成功，则网站可以获得用户信息并在页面上显示验证结果。算法 4 中显示了详细信息。

Algorithm 4. Verify identity algorithm
算法 4. 身份识别算法

```
function VerifyIdentity (D8, D10, PKu, T3)
    T4 = time()
    if (T4 − T3 > 10) then
        return false
    D11 = H(D1||D8||T3)
    if (D11! = D10) then
        return false
    if (Verify(PKu, D10) == false) then
        return false
    D12 = IPFSRequest(D8)
    D13 = Decrypt(SKw, D12)
    return D13
```

4. 实验与分析

4.1. 攻击模型

根据文献[10]的安全实验，我们建立了多个攻击模型实验来验证我们提出的机制。该模型包含“重放攻击模型”、“模拟攻击模块”和“网络钓鱼攻击模块”。我们将本机制与 OAuth2.0、OpenID [14]和 SAML2.0 [15]进行比较，以显示其安全性。各个攻击模型是基于以下条件进行假设构建的：

1) 攻击者可以随意加入本区块链网络。
2) 攻击者可以知道哪个 Whisper 请求是属于身份识别的。
3) 攻击者可以通过 Whisper 随意广播。
4) 攻击者可以拦截网络中的数据包。
5) 受害者无法区分钓鱼网站。

4.1.1. 重放攻击模型

当用户网站登录页并打开“LoginAuth”应用程序进行身份识别时，攻击者窃听了来自应用程序的请求，并窃取了 Whipsper 的请求包；然后攻击者用偷来的 Whisper 请求包使用受害者的会话登录网站并尝试窃取受害者的私人信息，如图 3。

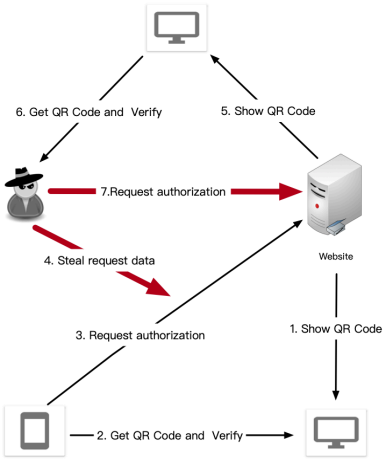

Figure 3. Reply attack module
图 3. 重放攻击模型

4.1.2. 钓鱼攻击模型

网络钓鱼攻击试图获取敏感的机密信息，例如用户名，密码，访问代码和其他安全信息。攻击者创建网络钓鱼网站，然后将 QR 码从合法网站转发给受害者。最后，攻击者开始从“LoginAuth”应用程序监听 Whisper 消息包以窃取受害者的私人信息，如图 4。

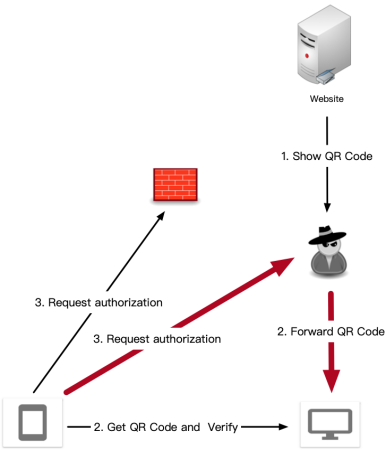

Figure 4. Phishing attack module
图 4. 钓鱼攻击模型

4.1.3. 模拟攻击模型

模拟攻击也称为欺骗攻击。在本机制中，“LoginAuth”应用程序将使用 Whisper 协议广播用户请求授权，攻击者可以通过拦截此请求并与网站建立会话，攻击者再修改了 Whisper 包的内容，将 topic 参数更改为自己的 topic 参数，最后广播了被篡改的 Whisper 包，如图 5。

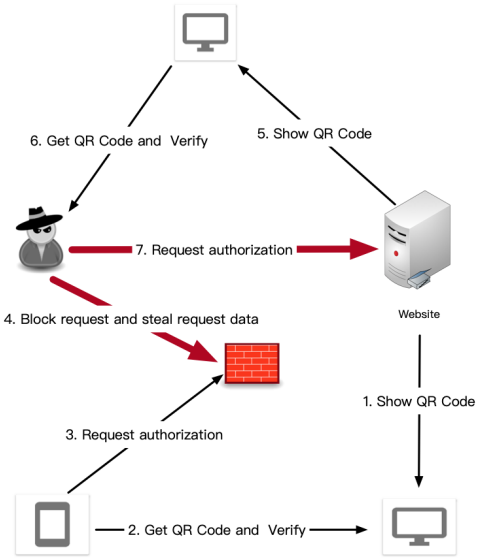

Figure 5. Impresonation attack module
图 5. 模拟攻击模型

4.2. 分析

4.2.1. 重放攻击漏洞

攻击者使用被盗的请求包进行身份识别请求，当网站接收到请求的数据，并验证数据签名也是有效的，但在防御此攻击的机制上，网站会将 whisper 的主题参数绑定到用户会话中，不同的会话拥有不同的主题参数，所以无法成功通过重放攻击的请求来进行身份识别。我们建议使用本机制的网站将 topic 参数设置为只能使用一次，并且有效期较短，这会使该机制更安全。遵循相同的防御策略，OAuth2.0、OpenID 和 SAML 请求参数绑定到用户会话也可以防御重放攻击[16]。

4.2.2. 钓鱼攻击漏洞

根据此机制，“LoginAuth”应用程序扫描 QR 码以获取网站信息，并使用 Whisper 协议进行身份识别请求。攻击者窃取了受害者要求进行身份授权的数据，但是由于攻击者没有合法网站的私钥，因此无法解密密文。这种机制可以防御网络钓鱼攻击。对于 OAuth2.0、OpenID 和 SAML，一般比较容易向服务提供商申请应用 ID，以假装它是合法的第三方企业，攻击者可以将此应用 ID 用于网络钓鱼站点，网络钓鱼网站成功地伪装成合法网站。受害者假如无法区分钓鱼网站的话，那么受害者就会被攻击成功。 所以 OAuth2.0、OpenID 和 SAML 无法防御类似这样的网络钓鱼攻击。

4.2.3. 模拟攻击漏洞

攻击者成功窃取了受害者的授权请求，然后篡改了 Whisper 的主题。攻击者广播了篡改后的 Whisper 的请求包，当网站收到了 Whisper 身份识别请求时，因为根据算法 4，D1 (topic)已被修改，则 D11 和 D12 不相等，所以验证无法成功。OAuth2.0 和 OpenID 可以通过比较状态字段来防御模拟攻击。对于 SAML，数据是在加密后传输的，难以实施攻击。

表 2 总结了此机制的实验结果，OAuth2.0、OpenID 和 SAML 在针对重放攻击，网络钓鱼攻击和模拟攻击在防御中的比较。为了提高此机制的安全性，Whisper 每个包的有效期应该尽可能短，例如 10 秒。Whisper 信封的主题参数必须至少为 64 个字符，并且为随机字符串，这使攻击者很难知道主题参数。区块链中的节点还可以不规则地广播一些无效的 Whisper 包，这可能会干扰攻击者的攻击，如表 2。

Table 2. Defense attack results
表 2. 防御对比

	Our mechanism	OAuth2.0	OpenID	SAML2.0
Replay attack	√	√	√	√
Phishing attack	√	×	×	×
Impersonation attack	√	√	√	√

5. 总结

在本文中，我们介绍了一种基于区块链的身份授权机制，并且我们的实验也证明了我们的机制在生产环境的可行性。使用此机制的网站不再需要实现类似 OAuth2.0 的回调机制，并且也能很好防御“重放攻击”，“网络钓鱼攻击”和“模拟攻击”。用户的信息被加密并存储在去中心化的文件系统(IPFS)中，以确保信息存储是去中心化的，没有单点漏洞。此机制可用于确保在不同节点之间通信具有可用性，机密性和隐私性，可应用于对网络身份识别具有更高要求的身份识别系统中；在之后，我们将继续基于 Whisper 协议在即时通讯安全方面的研究，并和现有的安全传输协议 TLS 进行对比，让信息在网络传输中更加安全和可靠。

参考文献

[1] Bosnjak, L. (2018) Brute-Force and Dictionary Attack on Hashed Real-World Passwords. 41*st International Convention on Information and Communication Technology, Electronics and Microelectronics* (*MIPRO*), Opatija, 21-25 May 2018, 1161-1166. https://doi.org/10.23919/MIPRO.2018.8400211

[2] Whisper Protocol. https://github.com/ethereum/wiki/wiki/Whisper

[3] Benet, J. (2014) IPFS-Content Addressed, Versioned, P2Ple System. https://arxiv.org/abs/1407.3561

[4] Wilkinson, S., Boshevski, T., Brandoff, J. and Buterin, V. (2018) Storj a Peer-to-Peer Cloud Storage Network. White Paper. https://storj.io/storj.pdf

[5] Vorick, D. and Champine, L. (2014) Sia: Simple Decentralized Storage. https://whitepaper.io/document/17/siacoin-whitepaper

[6] OAuth 2.0. https://oauth.net/2

[7] Sudhodanan, A., *et al*. (2018) Large-Scale Analysis & Detection of Authentication Cross-Site Request Forgeries. *IEEE European Symposium on Security and Privacy*, Paris, 26-28 April 2017, 350-365. https://doi.org/10.1109/EuroSP.2017.45

[8] Liu, W., Deng, X., Huang, G. and Fu, A.Y. (2006) An Anti-Phishing Strategy Based on Visual Similarity Assessment. *IEEE Internet Computing*, **10**, 58-65. https://doi.org/10.1109/MIC.2006.23

[9] Liu, W., Liu, G., Qiu, B. and Quan, X. (2012) Anti-Phishing through Phishing Target Discovery. *IEEE Internet Computing*, **16**, 52-61. https://doi.org/10.1109/MIC.2011.103

[10] Yang, F. and Manoharan, S. (2013) A Security Analysis of the OAuth Protocol. *Proceedings of Communications, Computers and Signal Processing*, Victoria, 27-29 August 2013, 271-276. https://doi.org/10.1109/PACRIM.2013.6625487

[11] Blockstack ID. https://blockstack.org

[12] uPort. https://www.uport.me

[13] Saritekin, R.A. (2018) Blockchain Based Secure Communication Application Proposal: Cryptouch. 2018 6*th International Symposium on Digital Forensic and Security* (*ISDFS*), Antalya, 22-25 March 2018, 1-4. https://doi.org/10.1109/ISDFS.2018.8355380

[14] OpenID. https://openid.net

[15] SAML2.0. https://wiki.oasis-open.org/security/FrontPage

[16] 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(4): 481-494.