華中科技大學

本科生毕业设计（论文）参考文献译文本

译文出处:

Daniel Fett. A Comprehensive Formal Security Analysis of OAuth 2.0. ACM Conference on Computer and Communications Security, 2016, 1204：1~12

院　　系　软件学院

专业班级　软件工程 1301

姓　　名　邢铭哲

学　　号　U201317429

指导教师　卢力

2017 年 1 月

译文要求

一、 译文内容须与课题（或专业内容）联系，并需在封面注明详细出处。

二、 出处格式为

图书：作者.书名.版本（第×版）.译者.出版地：出版者，出版年.起页～止页

期刊：作者.文章名称.期刊名称，年号，卷号（期号）：起页～止页

三、 译文不少于5000汉字（或2万印刷符）。

四、 翻译内容用五号宋体字编辑，采用A4号纸双面打印，封面与封底采用浅蓝色封面纸（卡纸）打印。要求内容明确，语句通顺。

五、 译文及其相应参考文献一起装订，顺序依次为封面、译文、文献。

六、 翻译应在第七学期完成。

译文评阅

导师评语

应根据学校“译文要求”，对学生译文翻译的准确性、翻译数量以及译文的文字表述情况等做具体的评价后，再评分。

评分：_____（百分制）　　指导教师(签名)：_____

2017 年 1 月 13　日

关于 OAuth2.0 的正式安全分析

摘要

OAuth 2.0 协议是最广泛部署的授权/单点登录（SSO）协议之一，也用作新的 SSO 标准 OpenID Connect 的基础。尽管 OAuth 的普及，到目前为止分析工作主要针对在特定实现中发现错误，并且基于正式模型从许多网络功能抽象，但是没有提供一个正式的解决。在本文中，我们进行了第一次广泛的正式分析在表达式网络模型中的 OAuth 2.0 标准。我们的分析旨在建立强大的授权，认证，和会话完整性保证，我们为此提供正式定义。在正式分析中，将涵盖所有四种 OAuth 授权类型（授权代码授予，隐式授权，资源所有者密码凭据授予和客户机凭证授予）。它们甚至可以在相同和不同的依赖方和身份提供方中同时运行，如恶意依赖方，身份提供商和浏览器。我们的建模和 OAuth 2.0 标准的分析假设遵循安全建议和最佳实践，以避免显而易见和已知的攻击。当在我们的模型中证明 OAuth 的安全性时，我们发现四次攻击，打破了 OAuth 的安全性。漏洞可以在实践中利用并且也存在于 OpenID Connect 中。我们建议修复所识别的漏洞，然后为第一次，实际上证明了 OAuth 的安全性 web 模型。特别是，我们展示了 OAuth 的固定版本，提供安全建议和最佳实践，提供了我们指定的授权，身份验证和会话完整性属性。

1. 介绍

OAuth 2.0 授权框架定义了一个基于 Web 的协议，允许用户授予网站访问它的资源（数据或服务）在其他网站（授权）。 前者网站称为依赖方（RP），后者被称为身份提供者（IdP）. 实际上，通常使用 OAuth 2.0 认证。也就是说，用户可以使用她在 RP 登录由 IdP（单点登录，SSO）管理的身份。授权和 SSO 解决方案已经在过去几年在网络中得到广泛采用，OAuth 2.0 是其中之一最流行的框架。 OAuth 2.0 在下文中通常简称为 OAuth，由身份提供者使用，例如 Amazon，Facebook，Google，Microsoft，Yahoo，GitHub，LinkedIn，StackExchange 和 Dropbox。这使数十亿用户可以登录数百万 RP 或与这些共享他们的数据，使 OAuth 成为网络上最常用的单点登录系统之一。OAuth 也是新的单点登录协议的基础 OpenID Connect，已在使用中并得到积极支持通过 PayPal（“使用 PayPal 登录”），还有 Google 和 Microsoft。考虑到 OpenID Connect 的广泛行业支持，在接下来的几年中，OpenID Connect 的广泛采用似乎成为可能。 OpenID Connect 基于 OAuth 构建，并清楚地提供用于用户认证的定义接口和附加（可选）功能，例如动态身份提供程序发现和依赖第三方注册，消息的签名和加密，以及注销。在 OAuth 中，用户与其浏览器之间的互动，RP 和 IdP 可以在四个不同的流中执行，或者授权类型：授权码授予，隐式授权，资源所有者密码凭据授予，以及客户端凭据授

予（我们参考以下作为模式）。此外，所有这些模式提供更多选择。这项工作的目标是提供一个深入的安全分析的 OAuth。分析 OAuth 的安全性是一项具有挑战性的任务，一方面由于 OAuth 的各种模式和选项提供，另一方面由于固有的复杂性网络。到目前为止，大多数关于 OAuth 安全性的分析是努力针对在特定实现中发现错误，而不是对综合分析标准本身。这可能是迄今为止对 OAuth 进行的最详细的正式分析中的一个。但是，没有一个现有的去分析 OAuth 帐户的所有 OAuth 的模式运行，这同时可能潜在地引入新的安全风险。事实上，许多现有方法仅分析授权代码模式和 OAuth 的隐式模式。另外，重要的是，没有基于全面的分析正式的网络模型（见下文），然而，这是必要的，以排除在上下文中运行协议时出现的安全风险的常见网络技术（更多详细信息，请参见第 6 节讨论相关工作）。

本文的贡献。我们对所有四种模式执行 OAuth 2.0 标准的第一次广泛正式分析，它们可以在同一个不同的 RP 和 IdP 中同时运行，基于一个全面的网络模型，覆盖大部分的现实世界设置中的浏览器和服务器交互。我们的分析还涵盖恶意 IdP，RP 和浏览器/用户的情况。我们对 OAuth 的正式分析使用 Fett，Küsters 和 Schmitz（FKS）提出的 Web 基础设施的表达式 Dolev-Yao 风格模型。 FKS 模型已被用于分析 Browser ID 单点登录系统的安全性以及 SPRESSO 单点登录系统的安全性和隐私性。此 Web 模型独立于特定 Web 应用程序设计，并严格模仿 Web 发布（事实上）的标准和规范，例如 HTTP / 1.1 和 HTML5 标准以及相关（建议）标准。它是迄今为止最全面的网络模型。其他人，HTTP（S）请求和响应，包括几个头像，如 cookie，位置，严格传输安全（STS）和原始头文件。 Web 浏览器的模型捕获了窗口，文档和 iframe 的概念，包括复杂的导航规则，以及新技术，如 web 存储和 web 消息（通过 post Message）。 JavaScript 通过所谓的脚本以抽象方式建模，可以发送，其中可以创建 iframe 并启动 XML HTTP Requests（XHR）。浏览器可能被手动态破坏。使用通用 FKS 模型，我们构建一个 OAuth 的正式模型，紧跟 OAuth 2.0 标准（RFC6749）。 自从 RFC 不修复协议的所有方面，为了避免已知的实现攻击，我们使用 OAuth 2.0 安全建议（RFC6819），附加 RFC 和 OAuth 工作组草案（例如 RFC7662）和当前 web best 实践（例如，关于会话处理）以获得具有现有的安全特征的 Auth 的模型，同时做出尽可能少的假设。此外，如上所述，我们的模型包括（同时）支持所有四种模式的 RP 和 IdP，并且可以被对手动态地破坏。此外，我们模拟 OAuth 的所有配置选项（参见第 2 节）。安全属性的规范化。基于此 OAuth 模型，我们提供 OAuth 的三个中央安全属性：授权，身份验证和会话完整性，其中会话完整性涉及授权和身份验证.Auth 2.0 对 OAuth 和修订。在试图证明这些属性时，我们发现了四次对 OAuth 的攻击。在第一次攻击，破坏授权和身份验证属性，IdP 无意中将用户凭据（即用户名和密

码）转发给 RP 或攻击者。在第二次攻击（IdP 混合）中，扮演 IdP 角色的网络攻击者可以冒充任何受害者。这种严重攻击再次打破了授权和身份验证属性，这是由 OAuth 2.0 协议中的逻辑缺陷引起的。两次进一步的攻击允许攻击者强制浏览器在 RP 下以攻击者的名字登录或强制 RP 使用攻击者的资源而不是用户的资源，打破了会话的完整性属性。我们已经验证了对 OAuth 和 OpenID Connect 的实际实现的所有四种攻击。我们在第 3 节中详细介绍了 OAuth 的攻击。在我们的技术报告中，我们将展示如何在 OpenID Connect 中利用这些攻击。Wealso 展示了如何通过在 OAuth 和 OpenID Connect 的新的和现有部署中容易实现的改变来修复攻击。我们通知各个工作组，他们确认了攻击，并且需要对标准/建议进行更改。IdP mix-up 攻击已经导致了一个新的 RFC 草案。对 OAuth 2.0 的正常分析。使用我们的 OAuth 模型和修复，我们能够证明 OAuth 所提到的满足安全的属性。这是第一个在一个综合性和表达性的网络模型中（见第 6 节）OAuth 的中心安全属性的证明。我们强调，如前所述，我们使用安全建议和最佳实践来模拟 OAuth。如第 5 节所讨论的，不遵循这些建议和最佳实践的实现可能容易受到攻击。事实上，在具体实现中的许多这样的攻击已经在文献中指出。因此，我们的结果还提供了针对安全 OAuth 实施的指南。此外，请注意，尽管这些结果为 OAuth 提供了强大的安全保证，但它们并不直接暗示 OpenID Connect 的安全性，因为 OpenID Connect 会在 OAuth 上添加特定详细信息。我们对 OpenID Connect 进行未来工作的形式分析。这里获得的结果可以作为这种分析的良好基础。

本文结构。在第 2 节中，我们使用授权代码模式作为示例提供 OAuth 2.0 的详细说明。在第 3 节，我们提出我们在分析过程中发现的攻击。 第 4 部分提供了我们在分析中构建的 FKS 模型的概述，第 5 部分提供了 OAuth 的正式分析。第 6 部分讨论了相关工作。第 7 部分讨论了全部细节，包括攻击如何应用于 OpenID Connect，我们的 OAuth 模型的更多细节和我们的安全证明，可以在我们的技术报告中找到。

2. OAUTH 2.0

在本节中，我们提供 OAuth 授权码模式的描述，其他三种模式仅作简要说明。在我们的技术报告中，我们提供了有关三种模式（授权类型）的详细描述。用于授权，即用户授权 RP 在 IdP 处访问用户数据（称为保护资源）。例如，用户可以使用 OAuth 来授权诸如 IFTTT3 的服务来在 Facebook 上访问她（私人）时间线。在这种情况下，IFTTT 是 RP 和 Facebook 的 IdP。概括地说，在最常见的模式中，OAuth 工作如下：如果用户想授权 RP 在 IdP 访问用户的一些数据，RP 重定向用户，用户的浏览器到 IdP，其中用户认证并同意允许 RP 访问 IdP 上的她的一些用户数据。然后，与由 IdP 发出的一些令牌（授权码或访问令牌）一起，用户

被重定向回 RP。然后 RP 可以使用这个令牌作为 IdP 的凭证来访问 IdP 的用户数据.OAuth 也通常用于认证，虽然它没有考虑到身份验证。例如，用户可以使用她的 Facebook 帐户（Facebook 是 IdP）在社交网络 Pinterest（RP）上登录。通常，为了登录，用户授权 RP 访问 IdP 处的唯一用户标识符。然后 RP 检索此标识符，并认为此用户已登录。

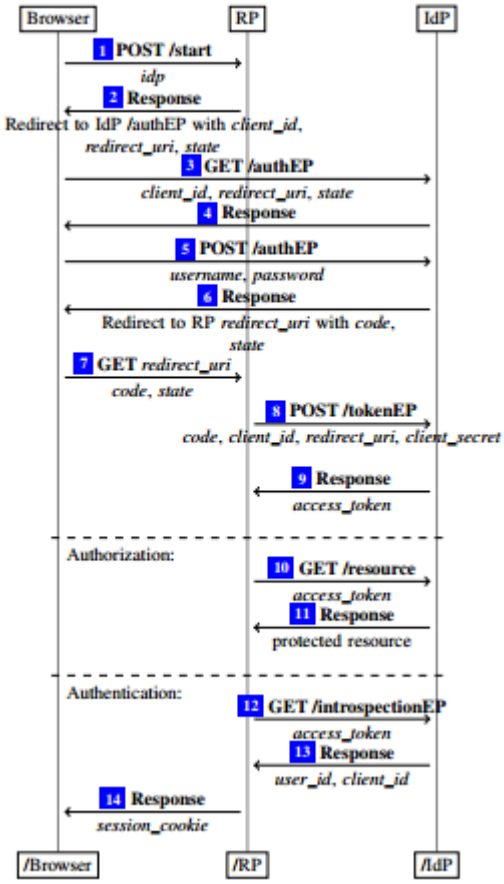

Figure 1 OAuth 2.0 **授权代码模式。 注意数据在如下所示箭头在** URI **参数，**HTTP **头或** POST **主体中传输。**

在 RP 可以与 IdP 交互之前，RP 需要在 IdP 上注册。注册过程的详细信息超出了 OAuth 协议的范围。在实践中，这个过程通常是手工任务。在注册过程中，IdP 向 RP 分配凭据：公共 OAuth 客户端标识和（可选）clientecret。 （回想一下，在 OAuth 标准的术语中，客户端“客户端”代表 RP）.RP 稍后可以使用客户机密钥（如果发出）来认证 IdP。另外，RP 注册一个或多个重定向端点 URI 在 RP）在 IdP。正如我们将在下面看到的，在一些 OAuth 模式中，IdP 将用户的浏览器重定向到这些 URI 之一。注意（取决于 IdP 的实现）RP 也可以将模式注册为重定向 URI，然后在 OAuth 运行。在所有模式下，OAuth 提供了几个选项，如上所述。为了简洁（并且与我们的分析相反），在以下描述中，我们仅考虑特定的选项集。例如，我们假设 RP 总是提供重定向 URI 并与 IdP 共享 OAuth 客户端密钥。

授权代码模式。当用户尝试授权 RP 在 IdP 访问她的数据或在 RP 登录时，RP 首先将用户的浏览器重定向到 IdP。然后，用户例如通过提供她的用户名和密码来认证 IdP，并且一起被重定向回到 RP 以及由 IdP 生成的授权码。 RP 现在可以使用此授权码（连同客户端标识和客户端密钥）联系 IdP，并接收访问令牌，RP 可以使用该访问令牌作为访问 IdP 的用户受保护资源的权限。逐步协议流。在下文中，我们逐步描述授权码模式的协议流程（也参见图 1）。首先，用户例如通过点击按钮来选择 IdP 来启动 OAuth 流，导致请求 1 被发送到 RP。 RP 选择其重定向端点 URI 之一 redirect_uri（其将在 7 中稍后使用）和值状态（其将用作防止 CSRF 攻击的令牌）。RP 然后将浏览器重定向到在 2 和 3 中的 IdP 处的所谓的授权端点 URI，其中它的 client_id，redirect_uri 和状态作为参数附加到 URI。然后，IdP 在 4 中提示用户提供其用户名和密码。用户的浏览器在 5 中将此信息发送到 IdP。如果凭证正确，IdP 创建一个 noncecode（授权代码），并将用户的浏览器重定向到 6 和 7 中的重定向端点 URI redirect_uri，并将代码和状态作为参数附加到 URI。如果 state 与上述相同，则 RP 在 8 中联系 IdP 并提供代码，client_id，client_secret 和 redirect_uri。然后，IdP 检查该信息是否正确，即检查该代码是否由 client_id 发出的 RP，client_secret 是 client_id 的秘密，redirect_uri 与步骤 2 中的秘密是否一致，以及该代码之前未被赎回。如果这些检查成功，IdP 在 9 中发出访问令牌 access_token。现在，RP 可以使用 access_token 在 IdP（授权）或登录用户（身份验证）访问用户的受保护资源，如下所述。当 OAuth 用于授权时，RP 使用 access_token 查看或为了认证操纵受保护资源，RP 使用接入令牌（步骤 12 和 13）获取用户 id（其唯一地标识在 IdP 处的用户）。然后，RP 向用户的（例如在步骤 10 和 11 中）发出会话 cookie。注意，为了使支持多个 IdP 的 RP 处理步骤 7，RP 必须知道用户想要用于授权的哪个 IdP。在实践中有两种不同的方法：第一，RP 可以使用不同的重定向 URI 来区分不同的 IdP。我们称之为 naïve user 意图跟踪。第二，RP 可以在步骤 1 之后的会话中存储用户意图，并且稍后使用该信息。我们称这种明确的用户意图跟踪。这同样适用于下面提供的 OAuth 的隐式模式。

隐式模式。该模式类似于授权 codemode，但是不是提供授权码，而是 IdP 通过用户的浏览器直接向 RP 发送访问令牌。更具体地，在隐式模式中，步骤 1-5（见图 1）与授权码模式相同。 IdP 不是创建授权码，而是立即发出访问令牌，并将用户的浏览器重定向到 RP 的重定向端点，并将 URI 的片段中包含的访问令牌。 （记住，碎片是由'#'符号指示的 URI 的特殊部分。）由于碎片不是在 HTTP 请求中发送的，所以当浏览器与 RP 联系时，访问令牌不会立即传送。在 RP 中，RP 需要使用 JavaScript 来检索片段的内容。通常，这样的 JavaScript

是在 RP 的回答中在重定向端点发送的。正如在授权码模式中一样，RP 现在可以使用访问令牌用于授权或认证（类似于图 1 的步骤 10 至 14）。

资源所有者密码凭据模式。 在此模式下，用户将其 IdP 的凭据直接提供给 RP。 RP 可以代表用户代表对 IdP 进行身份验证并检索访问令牌。 该模式旨在用于高度可信的 RP，例如用户设备的操作系统或高特权应用，或者如果先前的两种模式不可能执行（例如，对于没有 web 浏览器的应用）。

客户端凭据模式。与上面所示的模式相反，这种模式在没有用户交互的情况下工作。相反，它由 RP 启动以便获取访问令牌以访问 IdP 处的 RP 的资源。 例如，Facebook 允许 RP 使用客户端凭据模式获取访问令牌，以访问其广告效果的报告。

3. 攻击

如引言中所述，在试图证明基于 FKS Web 模型和我们的 OAuth 模型的 OAuth 的安全性时，我们发现了对 OAuth 的四次攻击，我们称之为 307 重定向攻击，IdP 混合攻击，状态泄漏攻击， 初始 RP 会话完整性攻击。 在本节中，我们提供这些攻击的详细描述以及可轻松实施的修复。 我们对 OAuth 的正式分析（见第 5 节）表明，这些修复足以建立 OAuth 的安全性。这些攻击也适用于 OpenID Connect（见第 3.5 节）。 图 2 提供了攻击的适用范围。 我们已经验证了我们对 OAuth 和 OpenID Connect 的实际实现的攻击，并向确认了攻击的各个工作组报告攻击（参见第 3.6 节）。

307 重定向攻击。在这种破坏我们的授权和身份验证属性（参见第 5.2 节）的攻击中，攻击者(运行恶意 RP)在用户使用错误的 HTTP 重定向状态代码登录 IdP 时学习用户的凭据。虽然攻击本身是基于一个简单的错误，据我们所知，这是这种攻击的第一个描述。

假设。主要假设是：（1）用于登录的 IdP 在将用户的浏览器重定向回 RP（图 1 中的步骤 6）时选择 307 HTTP 状态代码，以及（2）IdP 在用户的浏览器之后立即重定向用户用户输入的凭证（即，在对包含由用户的浏览器发送的表单数据的 HTTP POST 请求的响应中）。这个假设是合理的，因为 OAuth 标准和 OAuth 安全考虑（也不是 OpenID Connect 标准）都指定了如何重定向的确切方法。 OAuth 标准明确允许任何 HTTP 重定向：虽然本规范中的示例显示使用 HTTP 302 状态代码，但是允许通过用户代理实现此重定向的任何其他方法，并且被认为是实现细节。步骤 13 中的 IdP 包括 RP 的 OAuth client id，它在认证用户时由 RP 检查（参见 RFC7662）。此检查阻止在 OAuth 隐式模式下重新使用访问令牌跨 RP。授权时不需要此检查。假设（2）这个假设是合理的，因为在实践中可以发现在输入用户凭证之后立即重定向的许多示例，例如在 github.com（其中，不满足假设（1））。

攻击。当用户使用授权代码或隐式模式 OAuth 在恶意 RP 登录时，她被重定向到 IdP 并提示输入她的凭据。然后，IdP 在 POST 请求中从用户的浏览器接收凭据。 它检查凭据并将用户的浏览器重定向到响应 POST 请求的 RP 重定向端点。 自从 307 状态码用于此重定向，用户的浏览器将向包含来自先前请求的所有表单数据(包括用户凭据)的 RP 发送 POST 请求。由于 RP 由攻击者运行，他可以使用这些凭据来模拟用户。与 OAuth 标准中的当前字词相反，重定向的精确方法不是 OAuth 安全性的实现细节。在 HTTP 标准中，只有 303 重定向被明确定义为删除 HTTP POST 请求的主体。因此，OAuth 标准应该为上述步骤提供 303 重定向，以解决此问题。

攻击授权代码模式。我们现在描述对 OAuth 授权码模式的 IdPMix-Up 攻击。如上所述，非常类似的攻击也适用于隐式模式。如果 IdP 仅支持这两种模式中的一种，则两种攻击也起作用。用于授权码模式的 IdP 混合攻击在图 3 中描述。与在常规流中一样，当用户选择她想要登录时，攻击开始 HIdP（图 3 中的步骤 1)。现在，攻击者拦截预期用于 RP 的请求，并通过将 HIdP 替换为 AIDP8 来修改该请求的内容。攻击者再次拦截和修改 RP 3 的响应（包含重定向到 AIdP)，使得它重定向用户到 HIdP 4。攻击者还使用 HIdP（这是公共信息）处的 RP 的客户端 id 替换 AldP 上的 RP 的 OAuth client id。 （注意，我们假设从这一点开始，根据 OAuth 安全建议，用户的浏览器与 HIdP 和 RP 之间的通信使用 HTTPS 加密，因此不能被攻击者检查或更改)。然后用户认证到 HIdP 并且被重定向回到 RP 8。 RP 认为，由于攻击的第 2 步，此重定向中包含的 nonce 代码由 AlP 发出，而不是 HIdP。因此，RP 现在尝试在 AldP 10 而不是 HIdP 上为该访问令牌兑换此 nononce。这泄漏给攻击者。如果 HIdP 在注册期间没有向 RP 发出 OAuth 客户端返回，攻击者现在可以在 HIdP（11 和 12）中兑换访问令牌的代码.9 此访问令牌允许攻击者访问 HIdP 上的用户的受保护资源。这将打破授权属性(见第 5.2 节)。我们注意到，在这一点上，攻击者甚至可能提供虚假信息 8.在这一点上，攻击者还可以在 RP 读取用户会话的会话 ID。然而，我们的攻击不是基于这种可能性，并且即使 RP 在用户登录并且连接受 HTTPS 保护（会话管理的最佳实践）时更改此会话 ID 也是工作。在 RP 必须提供客户机密钥，这不会工作在这种模式（另请参见图 2)。回想一下，在这种模式下，客户机秘密是可选的。对于 RP，用户或其受保护的资源：他可以发出自创建的访问令牌，然后 RP 将使用该令牌访问攻击者的这种信息。为了破坏认证属性（见第 5.2 节）并假冒诚实的用户,攻击者在步骤 10 中获得代码后,在 RP 启动一个新的登录过程(使用他自己的浏览器)。他选择 HIdP 作为此登录过程的 IdP，并接收重定向到 HIdP，他忽略。这个重定向包含一个新的登录会话的 cookie 和一个新的状态参数。攻击者现在发送代码到 RP 模仿一个真正的登

录（使用 cookie 和新的状态值从上一个响应）。然后，RP 在 HIdP 使用代码检索访问令牌，并使用此访问令牌获取（诚实）用户的 ID。由于相信攻击者拥有诚实的用户帐户，RP 会向攻击者发出一个 session cookie 用于此帐户。结果，攻击者在 RP 下以诚实用户的 ID 登录。（请注意，在这种情况下，入侵者不会学习访问令牌。）

4. 总结

在本文中，我们基于一个全面和富有表现力的 web 模型，进行了第一次广泛的 OAuth 2.0 的正式分析。我们的分析，针对标准本身，而是特定的 OAuth 实现和部署，包括 OAuth 和可用选项的所有模式（授权类型），并且也采取恶意 RP 和 IdP 以及损坏的浏览器/用户计数。我们的 OAuth 模型和其分析的通用网络模型是迄今为止最全面的网络模型。我们的深入分析揭示了对 OAuth 的四个攻击以及基于 OAuth 的 OpenID 连接。我们验证了攻击，建议的修复程序，并向 OAuth 和 OpenID Connect 的工作组报告了攻击和我们的修复。工作组确认了攻击。修正标准和建议目前正在讨论中或已经纳入新 RFC 的草案。通过应用修复，我们能够证明 OAuth 2.0 的强大的授权，身份验证和会话完整性属性。我们的安全分析假设 OAuth 安全建议和某些最佳实践。我们显示，否则 OAuth 的安全无法保证。通过这一点，我们还提供了实施的明确指导。 OAuth 是 Web 中最广泛部署的授权和身份验证系统之一，也是其他协议的基础，这使我们的分析具有相关性。对于未来的工作，我们对 OAuth 的正式分析为 OpenID Connect 的正式分析提供了一个良好的开端，因此，这样的分析是我们研究的明显的下一步。

5. 参考文献

[1] M. Abadi and C. Fournet. Mobile Values, New Names, andSecure Communication. In POPL 2001, pages 104–115.ACM Press, 2001.

[2] D. Akhawe, A. Barth, P. E. Lam, J. Mitchell, and D. Song.Towards a Formal Foundation of Web Security. In CSF 2010,pages 290–304. IEEE Computer Society, 2010.

[3] A. Armando, R. Carbone, L. Compagna, J. Cuéllar,G. Pellegrino, and A. Sorniotti. An authentication flaw inbrowser-based Single Sign-On protocols: Impact andremediations. Computers & Security, 33:41–58, 2013.Elsevier, 2013.

[4] A. Armando, R. Carbone, L. Compagna, J. Cuéllar, and M. L.Tobarra. Formal Analysis of SAML 2.0 Web Browser SingleSign-on: Breaking the SAML-based Single Sign-on forGoogle Apps. In FMSE 2008, pages 1–10. ACM, 2008.

[5] C. Bansal, K. Bhargavan, A. Delignat-Lavaud, and S. Maffeis.Keys to the Cloud: Formal Analysis and Concrete Attacks onEncrypted Web Storage. In POST 2013, volume

7796 ofLNCS, pages 126－146. Springer, 2013.

[6] C. Bansal, K. Bhargavan, A. Delignat-Lavaud, and S. Maffeis.Discovering Concrete Attacks on Website Authorization byFormal Analysis. Journal of Computer Security,22(4):601－657, 2014. IOS Press, 2014.

[7] A. Barth, C. Jackson, and J. C. Mitchell. Robust defenses forcross-site request forgery. In CCS 2008, pages 75－88. ACM,2008