InviCloak: An End-to-End Approach to Privacy and Performance

贾怀宇2020090916007

论文原稿：InviCloak: An End-to-End Approach to Privacy and Performance [1]

<https://arxiv.org/pdf/2209.01541.pdf>

收录于 CCS ’22, November 7–11, 2022, Los Angeles, CA, USA

InviCloak: 应对隐私和性能的端到端方法

**1.摘要：**

在当今的网络生态系统中，使用内容交付网络（CDN）的网站与CDN共享其传输层安全（TLS）私钥或会话密钥。在本文中，我们介绍了一篇来自CCS2022的文章：InviCloak的设计和实现，该系统可以在不改变TLS或升级CDN的情况下保护用户和网站私人通信的保密性和完整性。InviCloak建立了一个轻量级但安全实用的密钥分发机制，使用现有的DNS基础设施来分发与网站域名相关的新公钥。网络客户端和网站可以使用新的密钥对在TLS内建立一个加密通道。InviCloak适应了当前的网络生态系统。网站可以在没有客户参与的情况下单方面部署InviCloak，以防止CDN内部的被动攻击者窃听其通信。如果客户端也安装了InviCloak的浏览器扩展，那么在CDN内的主动攻击者存在的情况下，客户端和网站可以实现端到端的保密和不受干扰的通信。除此之外，我们还介绍了原文中提到的其他CDN，以此来和InviCloak做对比。

Keywords：CDN; HTTPS; Private key sharing

**2.引言：**

**2.1 概述**

内容交付网络（CDN）在网络生态系统中发挥着重要作用。它们不仅加快了网络内容的传播速度，而且还保护网站免受各种攻击。例如，Akamai、Cloudflare和CloudFront等CDN提供分布式拒绝服务（DDoS）攻击缓解和恶意内容清除服务。[2]

在本文设计的核心中，旨在探索一种具有不同成本和效益权衡的解决方案，供市场选择。该解决方案，InviCloak，采取了一种端到端的方法。它适应了当前的密钥共享做法，并将内容服务授权与保密性分开。一个网站使用共享的TLS密钥来授权CDN提供其非隐私敏感的内容。然后，它使用一对新的私人/公共密钥，不与CDN共享，以保护隐私敏感的内容。InviCloak可以防止主动攻击，并且不增加发送到源服务器的流量。

InviCloak使用现有的DNSSEC和DNS-over-HTTPS（DoH）基础设施来分发网站的新公钥，从而避免了网站为新域名获取新的TLS证书。为了便于部署，InviCloak在网络客户端和网站源服务器之间的现有TLS会话中嵌入了一条端对端加密隧道，以传输私人数据，如用户的登录密码。

**2.2 模型优点**

（1）不改变底层TLS协议，对CDN完全透明。

（2）它不修改现有的网络服务器实现，网络开发人员不需要改变现有的网络资源或管理新的域名和证书。

（3）InviCloak使用一对新的私人/公共密钥，不与CDN共享，以保护隐私敏感的内容。

（4）InviCloak可以防止主动攻击，并且不增加发送到源服务器的流量。

**3.知识补充：**

**3.1 CDN**

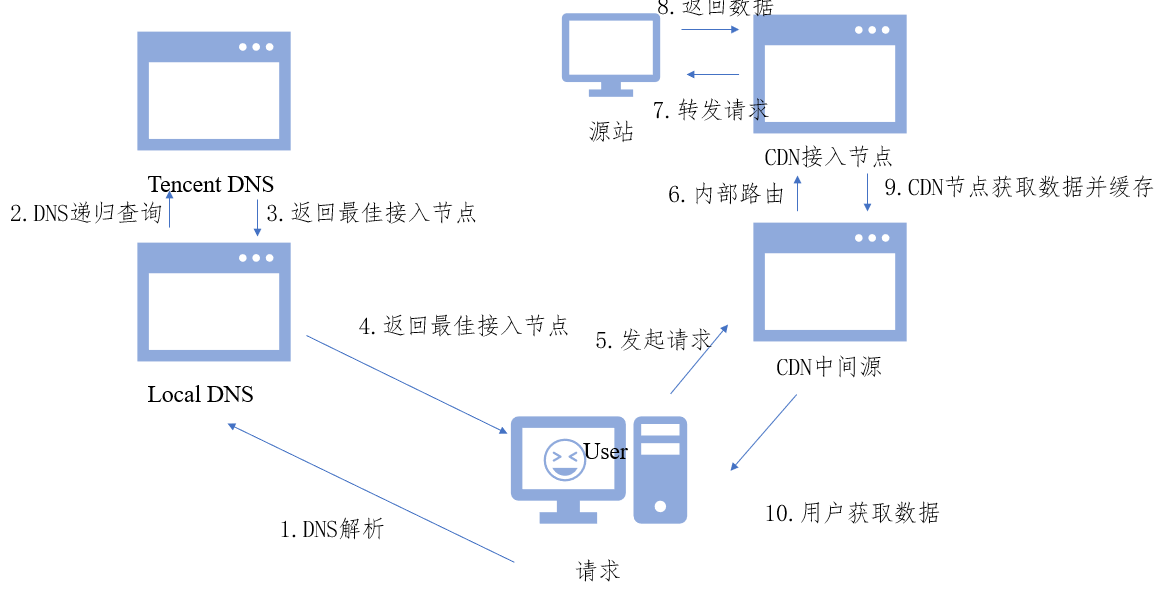
内容分发网络（英语：Content Delivery Network或Content Distribution Network，缩写：CDN）是指一种透过互联网互相连接的电脑网络系统，利用最靠近每位用户的服务器，更快、更可靠地将音乐、图片、视频、应用程序及其他文件发送给用户，来提供高性能、可扩展性及低成本的网络内容传递给用户。

CDN工作流程：

1. 当用户访问已经加入CDN服务的网站时，首先通过DNS重定向技术确定最接近用户的最佳CDN节点，同时将用户的请求指向该节点。
2. 当用户的请求到达指定节点时，CDN的服务器（节点上的高速缓存）负责将用户请求的内容提供给用户。
3. 具体流程为: 用户在自己的浏览器中输入要访问的网站的域名，浏览器向本地DNS请求对该域名的解析，本地DNS将请求发到网站的主DNS，主DNS根据一系列的策略确定当时最适当的CDN节点，并将解析的结果（IP地址）发给用户，用户向给定的CDN节点请求相应网站的内容。

CDN的优势：

（1）CDN节点解决了跨运营商和跨地域访问的问题，访问延时大大降低。

（2）大部分请求在CDN边缘节点完成，CDN起到了分流作用，减轻了负载。

图一：CDN例子

用户向 www.test.com 下的某图片资源，如 1.jpg 发起请求，先要向 Local DNS 发起域名解析请求；

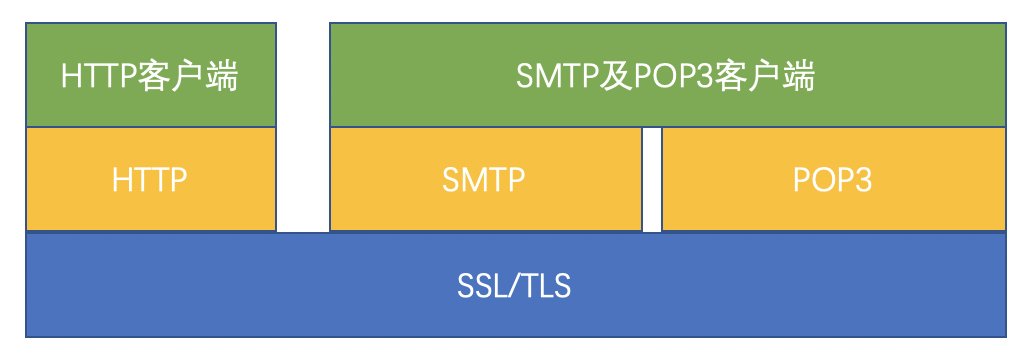
当 Local DNS 解析 www.test.com 时，会发现已经配置了 CNAME www.test.com.cdn.dnsv1.com，解析请求会发送至 Tencent DNS（GSLB），GSLB 为腾讯云自主研发的调度体系，会为请求分配最佳节点 IP；Local DNS 获取 Tencent DNS 返回的解析 IP；用户测获取解析 IP；用户向获取的 IP 发起对资源 1.jpg 的访问请求；若该 IP 对应的节点缓存有 1.jpg，则会将数据直接返回给用户（10），此时请求结束。若该节点未缓存 1.jpg，则节点会向业务源站发起对 1.jpg 的请求（6、7、8），获取资源后，结合用户自定义配置的缓存策略（可参考用户指南中缓存时间设置章节内容），将资源存储（9），并返回给用户（10），此时请求结束。

**3.2 TLS**

SSL/TLS是一种密码通信框架，他是世界上使用最广泛的密码通信方法。SSL/TLS综合运用了密码学中的对称密码，消息认证码，公钥密码，数字签名，伪随机数生成器等，可以说是密码学中的集大成者。

SSL(Secure Socket Layer)安全套接层，是1994年由Netscape公司设计的一套协议，并与1995年发布了3.0版本。

TLS(Transport Layer Security)传输层安全是IETF在SSL3.0基础上设计的协议，实际上相当于SSL的后续版本。



图二：TLS协议

TLS主要分为两层，底层的是TLS记录协议，主要负责使用对称密码对消息进行加密。

上层的是TLS握手协议，主要分为握手协议，密码规格变更协议和应用数据协议4个部分。

握手协议负责在客户端和服务器端商定密码算法和共享密钥，包括证书认证，是4个协议中最最复杂的部分。

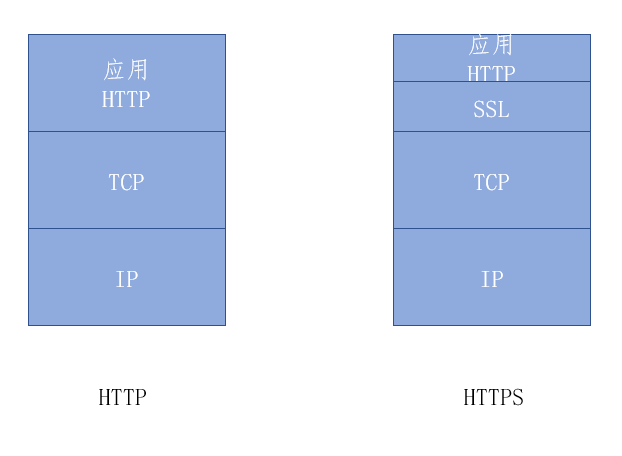
密码规格变更协议负责向通信对象传达变更密码方式的信号

警告协议负责在发生错误的时候将错误传达给对方

应用数据协议负责将TLS承载的应用数据传达给通信对象的协议。

**3.3 HTTPS**

HTTPS：是以安全为目标的HTTP通道，简单讲是HTTP的安全版，即HTTP下加入SSL层，HTTPS的安全基础是SSL，因此加密的详细内容就需要SSL。HTTPS协议的主要作用可以分为两种：一种是建立一个信息安全通道，来保证数据传输的安全；另一种就是确认网站的真实性。

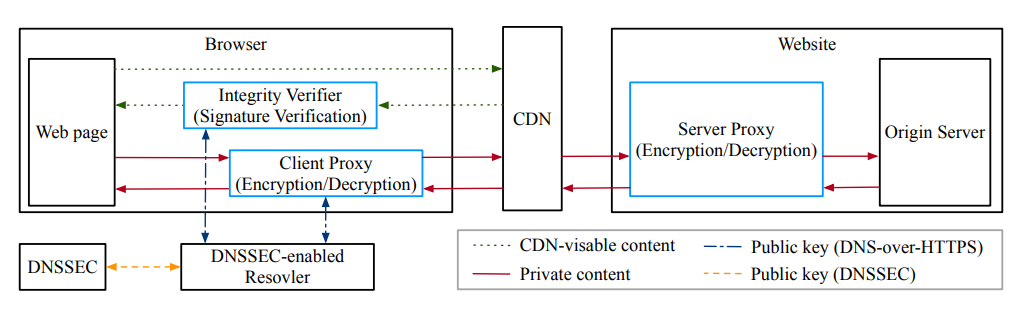


图三：HTTP与HTTPS

HTTP+加密+认证+完整性保护 = HTTPS

1. **模型与设计**

**4.1 模型**



图四：InviCloak模型

InviCloak引入了三个新的组件：一个在网络浏览器中运行的客户端代理，一个作为浏览器扩展引入的完整性验证器，以及一个在网站源服务器前作为反向代理运行的服务器代理。完整性验证器适用于所有支持InviCloak的网站，而客户端代理是一个JavaScript库，包括每个网站的特定配置。一个网站在其登陆页面中嵌入客户端代理，而完整性验证器将通过浏览者认可的机制（如扩展市场）分发。InviCloak使用DNS基础设施来分发网站的公钥，以便在浏览器和网站的源服务器之间建立一个加密通道。

* 1. **InviCloak的组件**

**Key Distribution & Management：**InviCloak使用现有的DNS基础设施来分发新的公钥。在InviCloak中，客户端代理和完整性验证器将在DNS-over-HTTPS（DoH）中向支持DNSSEC的解析器发送公钥查询。InviCloak在浏览器中缓存了一个网站的公钥，在用户清除缓存之前不会请求公钥。InviCloak在TLSA记录中存储公开密钥而不是证书。

**The Client Proxy**：在InviCloak的设计中，客户端代理是一个在网络浏览器中运行的JavaScript库。本文设计的客户端代理将配置文件作为输入，这样每个网站只需要配置文件来部署InviCloak，不需要开发新的JavaScript代码

**The Server Proxy**：本文把服务器代理设计成网络服务器的附加模块，如NGINX。一个网站不需要修改其网络服务器的实现来部署InviCloak。网站在服务器代理的配置文件中列出私人内容的URLs(nginx.conf)中列出私有内容的URL，与配置客户端代理的方式类似。

**Establishing the Encryption Channel**：。根据TLS 1.3，InviCloak将在每次密钥交换后随机化一个32字节的会话ID并存储相应的会话密钥。会话ID被返回给客户端。当客户端代理向服务器代理发送加密的私人内容时，它将包括明文的会话ID。服务器代理通过会话ID检索会话密钥，如果没有找到相应的会话密钥，它将拒绝该请求。密钥交换过程可以按需进行，也可以异步进行。

**Using the Encryption Channel：**会话密钥设置完成后，客户端和服务器代理可以使用它来加密和解密浏览器和网站源服务器之间的私人通信。当浏览器发起一个URL在客户端代理的配置文件列表中的请求时，客户端代理会生成一个包括会话ID、序列号和请求内容的消息。会话ID唯一标识它与服务器代理共享的InviCloak会话，而序列号唯一标识该会话中的请求。客户端代理使用与服务器代理共享的会话密钥对序列号和请求内容进行加密，使用TLS 1.3批准的密码组。然后，它将加密的请求发送到CDN，CDN将把它转发到源服务器。InviCloak不改变TLS，因此底层TLS连接不知道客户端代理添加的额外加密，并将加密请求视为普通的HTTP请求。

**Integrity Verifier**：InviCloak使用完整性验证来抵御主动对手。与客户端代理不同，完整性验证器是对现有浏览器的一个扩展。它是一个独立于网络服务的组件，可以在网络会话之外安全地获得。

**Partial Deployment of Integrity Verifier：**在InviCloak的设计中，网站可以在没有用户参与的情况下部署服务器代理和客户端代理，以击败被动对手，但用户需要自己安装完整性验证器，以击败主动对手。

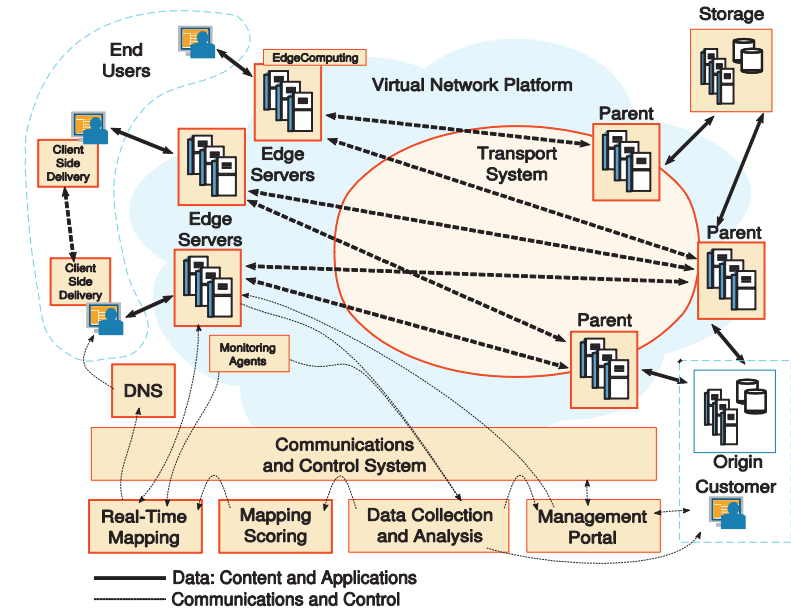
**Cookie Management**：InviCloak通过用客户端和服务器代理之间建立的会话密钥对用户的认证cookie进行加密来防止cookie泄漏。网站将在其服务器代理的配置文件中指定哪些cookie对CDN是私有的（称为私有cookie）。服务器代理将在HTTP响应的Set-Cookie标头中加密私人cookies，并在HTTP请求的Cookie标头中解密这些cookies。客户端代理不需要对cookie进行解密，因此浏览器将存储加密的cookie，并在需要时将其附加到请求头中。

1. **其余常见CDN**

**Akamai[4]**: Akamai维护着由24万多台“边缘服务器”组成的全球网络。 它们位于互联网的“边缘”，并尽可能靠近最终用户。 为了实现这一目标，许多边缘服务器甚至直接位于ISP或移动数据塔中，以进一步减少在移入Akamai网络之前连接到用户ISP的延迟。

当用户发出请求时，Akamai会将其动态映射到最近的可用边缘服务器。 边缘服务器在使用Akamai网络内所有其他边缘服务器之间的最佳可用路由以从源中获取内容之前，将应用内容提供商指定的业务规则。 业务规则在每个边缘服务器上复制。然后，将可用的任何配置为缓存的内容缓存在边缘服务器上，以供将来连接到该节点的请求使用。

例如：通过在DNS中添加一条CNAME记录将站点添加到Akamai，该记录从主机名(例如“ community.akamai.com”)指向Akamai控制的Akamai边缘主机名“ community.akamai.com.edgekey.net”。边缘服务器映射将接管以分配最佳可用边缘服务器。 如果您“ 挖掘 ”主机名并看到“ edgekey.net”，则说明内容提供商正在使用Akamai。



图五：Akamai平台

1. **总结**

本课程报告原论文是：《InviCloak: An End-to-End Approach to Privacy and Performancein Web Content Distribution》，核心观点是InviCloak，一个允许网站使用CDN进行DDoS保护和网络加速的系统，而不暴露它与用户交换的敏感数据。InviCloak对客户端和源服务器之间传输的敏感数据进行加密，这样，不被信任的CDN就无法窃听他们的通信。一个网站的单边部署可以防止CDN中的被动窃听者。如果用户安装了InviCloak的浏览器扩展，InviCloak可以防止CDN内的主动攻击者窃听或篡改其私人通信。此外，我们还罗列了论文中其余常见的CDN：Akamai，这是一个维护着由24万多台“边缘服务器”组成的全球网络。

参考文献

[1] Lin S, Xin R, Goel A, et al. InviCloak: An End-to-End Approach to Privacy and Performance in Web Content Distribution[C]//Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. 2022: 1947-1961.

[2] Omer Gil. 2017. Web Cache Deception Attack. https://omergil.blogspot.com/

2017/02/web-cache-deception-attack.html

[3] 2020. Let’s Encrypt Stats. <https://letsencrypt.org/stats/>

[4] Erik Nygren, Ramesh K Sitaraman, and Jennifer Sun. 2010. The Akamai Network:

a Platform for High-Performance Internet Applications. SIGOPS OSR 44, 3 (2010),

2–19.