ccTLD 的 DNSSEC 部署指南

ICANN 首席技术官办公室

亚兹德 • 阿坎诺 (Yazid Akanho) 和保罗 • 暮辰 (Paul Muchene)
OCTO-029
2021 年 11 月 12 日

ICANN

目录

本文是 ICANN 首席技术官办公室 (Office of the Chief Technical Officer, OCTO) 文档系列的一部分。要查看 OCTO 系列文档，请参阅 OCTO 出版物页面。关于这些文档，如果您有任何问题或建议，请将您的反馈发送至 octo@icann.org。

本文支持 ICANN 的战略目标，即通过与相关利益相关方合作加强域名系统 (Domain Name System, DNS) 的协调，改善维护 DNS 的安全和稳定的共同责任。加强 DNS 和 DNS 根服务器系统 (root server system, RSS) 的安全性是 ICANN 战略目标的一部分。

1 简介

近年来，安全日渐成为互联网的一大主要问题。特别是在域名系统 (DNS) 方面，多年来已提出并开发多种安全协议，域名系统安全扩展 (Domain Name System Security Extensions, DNSSEC) 是其中最重要的协议之一。DNSSEC 通过提供数据来源验证和数据完整性保护，帮助确保 DNS 响应的安全。

早在 2010 年 7 月，我们已通过 DNSSEC 对 ICANN 管理的 DNS 根区首先进行了签名。在本指南发布时，所有通用顶级域 (gTLD) 都将进行 DNSSEC 签名，一方面是出于其与 ICANN 的合同义务；但另一方面，只有约 60% 的国家和地区顶级域 (ccTLD) 已签名。ccTLD 层面出现这种趋势的原因之一可能是，ccTLD 经理人对于使用 DNSSEC 保护其区安全的过程缺乏相关认识。

因此，ICANN 首席技术官办公室 (OCTO) 发布了本指南，以利于 ccTLD 注册管理运行机构更好地掌握这一流程，从而帮助他们使用 DNSSEC 对所属区进行签名。本指南不涉及 DNSSEC 的第二个方面，即主要在通常位于互联网服务提供商 (Internet Service Providers, ISP)、大型公共云运营商或企业网络中的 DNS 递归解析器上进行的验证。

1.1 本文的目标受众

本指南的主要目的是向 ccTLD 注册管理机构经理人、工作人员、利益相关方（尤其是注册服务机构、注册人）以及任何其他人员简要介绍 DNSSEC，以及注册管理机构如何在区签名中实施 DNSSEC。本文不涉及技术配置细节；相反，它是一个比较基础的指南，使大家可以对 DNSSEC 协议、先决条件和签署 ccTLD 区时应考虑的部署注意事项有一个基本了解。

即使您运营的 TLD 已经使用 DNSSEC 进行了签名，本文也可以帮助您确定是否有需要改进之处，例如当前的算法是否遵循最佳实践，或者整个 DNS 服务的文档记录是否恰当。如果您是 ccTLD 运营商，或管理着已签名 ccTLD 下的区，那么本指南可以帮助您入门。尽管所有 gTLD 运营商都已签署各自的区，但他们也可以从本文所述关于 DNSSEC 运营最佳实践的见解中获益，并以此为基础，来提高其注册服务机构和注册人对 DNSSEC 的认识。

关于 DNSSEC 的理论、技术和运作有大量的文件，本指南引用了其中一些作为参考。因此，如果读者希望加深对 DNSSEC 某个或某些方面的知识或了解，欢迎阅读这些参考文件。

2 DNSSEC 以及它对 DNS 的价值

域名系统 (DNS) 是一个分层级、分布式、去中心化的互联网命名系统。与电话簿将名称转换为电话号码类似，DNS 可帮助将域名信息转换为 IP 地址，反之亦然。DNS 被认为是互联网上的一项关键服务，但它最初并没有设计强大的安全机制来保障其数据的完整性和真实性。多年来已发现许多威胁 DNS 可靠性和可信度的漏洞，在 DNSSEC 的帮助下，其中一些漏洞得以解决。

DNSSEC 的相关定义和说明主要出现在三个互联网标准文件中：RFC 4033，*DNS Security Introduction and Requirements（DNS 安全性介绍和要求）*；RFC 4034，*Resource Records for the DNS Security Extensions（DNS 安全扩展的资源记录）*；以及 RFC 4035，*Protocol*

Modifications for the DNS Security Extensions（DNS 安全扩展的协议修订）。DNSSEC 使用公共密钥加密（生成公私密钥对），将数据来源验证、数据完整性、核实和否定存在验证等功能添加到 DNS 之中。具体而言，它向 DNS 引入了数字签名和一组新的资源记录类型及消息头位（标志），以用于验证来自签名区的 DNS 响应。需要注意的是，DNSSEC 不加密任何 DNS 消息数据，因此不提供机密性。

使用 DNSSEC 对域进行签名之后，区管理员将使用私钥生成数字签名，并将其作为“资源记录数字签名 (Resource Record Signature, RRSIG)”记录发布到区文件中，作为该域的区数据的一部分。当安全感知递归解析器（也称为验证解析器）向已签名域的权威服务器发送一个 DNS 查询时，DNS 响应将包含明文或未加密格式的资源记录以及与之相关的数字签名。然后，解析器使用它收到的数字签名来验证此 DNS 响应。为此，验证解析器还会请求其他 DNSSEC 相关信息，例如存储在 DNSKEY 记录中并由域管理员在区数据中发布的公钥。

3 DNSSEC 部署的先决条件和要求

3.1 现有系统的文件

由于 DNS 在互联网上扮演着极为重要的角色，需要做到在任何情况下都不能中断服务，因此有必要维护一份对 DNS 基础设施、运营和流程进行描述的最新文档。另一方面，DNSSEC 为现有的 DNS 基础设施和运营又额外增加了一定的复杂性。因此，保持文档处于最新状态对于清晰掌握现有系统情况以供参考至关重要。此举在员工流动或基础设施升级时，还可保证运营的连续性。

我们建议这份文档主要概述两部分内容：ccTLD 的治理政策，以及这项服务的运营和技术方面。

此外，还建议这份文档包含尽可能多的信息，但不能包含任何可能会被利用对注册管理机构进行攻击的敏感或机密数据，如用户名和密码。

这份文档的治理可涵盖以下主题：

- ccTLD 的概述和结构
- 注册模式：3R（注册管理机构、注册服务机构和注册人）、2R（仅注册管理机构和注册服务机构）或其他模式
- 注册管理机构的技术和管理联系人
- 人力资源、角色、责任和参与注册管理机构技术决策过程的联系人
- 注册服务机构名单及相应的联系信息

文档在技术和运营方面可涵盖以下主题：

- 一级和二级权威域名服务器 (name servers, NS) 的数量及相应的 IP 地址；TLD 联系信息（电话和电子邮件地址）；注册管理机构和注册服务机构之间使用的协议，如可扩展供应协议 (Extensible Provisioning Protocol, EPP)；用于实现注册管理机构职能的软件和硬件，包括注册管理机构数据库、注册数据访问协议 (Registration Data Access Protocol, RDAP) 和/或 WHOIS 服务器，以及其他技术信息
- 用户访问和权限清单只能严格地提供给有限的授权人员

- 备份和恢复程序
- **安全性**：**物理**访问、日志管理、访问控制、密码管理、防火墙、区文件完整性和区传输安全性等等。
- 监控系统：硬件、软件、区同步（NS 之间）
- 维护策略
- 业务连续性和灾难恢复计划

3.2 审核现有基础设施

如前所述，部署 DNSSEC 会增加现有 DNS 基础设施和运营的复杂性。因此，一种好的安全做法是，由具备必要技能和自主权的外部或内部相关方对当前系统基础设施、运营和流程进行审计，识别并分享他们在审计结果和任何缺漏。在对区进行签名之前或在签名的过程当中，必须修复当前系统中的任何缺陷。此举可以降低在实施 DNSSEC 后事情变得脱离掌控的风险。

3.3 编写 DNSSEC 政策和实践声明

3.3.1 什么是 DNSSEC 政策和 DNSSEC 实践声明？

在对域进行签名时，需要考虑几个因素并定义一些参数，例如 DNSSEC 签名算法、密钥大小、签名有效期和签名刷新频率。在一个有众多授权的区，较好的做法是完整记录并及时更新适用于该区的一组参数，并公开提供这些参数。因此，这里应考虑两个概念：

- **DNSSEC 政策** (**DNSSEC Policy**, DP)：阐述要对 DNSSEC 签名区实施的安全要求和标准。DP **构成**对实体，如注册管理机构，进行审计、认证或评估的基础。可以根据每个实体要实施**的一个或多个** DP 对其进行评估。简言之，DP **指出需要完成哪些工作**。
- **DNSSEC 实践声明** (**DNSSEC Practice Statement**, DPS)：这是一份运营实践披露文件，可作为对 DNSSEC **政策（如果存在）的支持和**补充。此文件简要说明区运营商及其区管理合作伙伴（如有）**如何**实施相关流程和控制，以满足适用 DP **的要求。与 DP 相反**，DPS **指出**实际正在做的工作。

DP 提供一般性原则，而 DPS 提供对相关流程和控制的描述，因此比 DP 要更详细。另一方面，政策通常由政策权威机构（TLD 经理人或监管机构）编写，并且可能适用于 DNS 层级结构中的一个或多个区；而特定于单区的实践声明则由区运营商负责编写，描述其如何满足一项或一组具体政策的要求。

例如，在 ccTLD 的管理联系人和技术联系人是不同实体的情况下，管理联系人可以发布一份政策来概述要遵循的标准和要求，同时还可要求技术联系人发布一份实践声明，详细说明如何满足这些标准和要求。

或者，不受任何外部政策管辖的区运营商或经理人也可以发布 DPS。

如果实体运营的区包含大量授权（如 TLD），那么发布 DPS 对这样的实体来说很有意义。发布 DPS 有助于提高透明度，增加社群对 TLD 运营的信任，但如前所述，DPS 不应包含敏感运营信息。

RFC 6841，*A Framework for DNSSEC Policies and DNSSEC Practice Statements（DNSSEC 政策和 DNSSEC 实践声明的框架）*，该文档深入介绍了 TLD 运营商在分别定义 DP 和 DPS 时，应全面考虑哪些主题。

3.3.2 如何编写 DP 和 DPS

编写 DPS 是对 ccTLD 进行签名过程中的重要一步。DPS 可以短小精悍，或详尽复杂，但无论如何，它应帮助人们了解 DNSSEC 运营框架，以及人们如何能够信任 ccTLD 签名过程。

下表汇总了 RFC 6841 中概述的一组条款，由八个部分组成，在起草 DP 或 DPS 时可以考虑这些条款。并非 RFC 6841 中的所有部分都要求强制实施，因此您可以自由选择适合您需要的（子）条款。

标题	描述	子条款
简介	确定和介绍一组条款，并指出该政策或实践声明适用于哪些类型的实体和应用。	• 概述 • 文件名称和标识 • 社群和适用性 • 规范管理
发布和存储库	描述实体在发布有关其实践、公钥、此类密钥当前状态等信息时需要遵循的要求，以及与保存这些信息的存储库相关的详细信息。	• 存储库 • 公钥发布
运营要求	描述运营一个 DNSSEC 签名区时的运营要求。	• 域名的含义 • 子区经理人的标识和身份验证 • DS 资源记录的注册 • 用于证明私钥的拥有和所有权的方法 • DS 资源记录的移除

设施、管理和运营控制	描述非技术方面的安全控制，即对物理、流程和人员的控制，以便安全执行 DNSSEC 相关职能。此类控制包括物理访问、密钥管理、灾难恢复、审计和归档。这些非技术方面的安全控制对于信任 DNSSEC 生成的签名至关重要。	• 物理控制 • 流程控制 • 人员控制 • 审计日志记录流程 • 破坏和灾难恢复 • 实体终止
技术安全控制	定义为管理与 DNSSEC 运营相关的加密密钥和激活数据（例如，PIN 码、密码或手动持有的密钥共享）而采取的安全措施。	• 密钥对的生成和安装 • 私钥保护和加密模块工程控制 • 激活数据
区签名	涵盖区签名的所有方面，包括围绕签名密钥的加密规范、签名机制、密钥轮转方法和实际的区签名。 子区和其他依赖方可依据本节中的信息来了解签名区中应有的数据，并确定其自身行为。	• 密钥长度、密钥类型和算法 • 否定存在验证 • 签名格式 • 密钥轮转 • 签名生命周期和重新签名频率
合规审计	描述区运营商以及其他可能的相关实体如何进行审计。	• 实体合规性审计频率 • 审计人员的身份/资格 • 审计涉及的主题 • 审计后的行动
法律事项	指明注册管理机构在哪个司法管辖区内运营，并指出任何有效力的相关协议。 “法律事项”部分可告知已确定会对保护个人身份隐私信息产生的任何影响。	• 提及适用的司法管辖权 • 合同义务和遵守的国家/地区、跨国或国际法律法规 • 个人身份隐私信息的数据保护和处理

可在此框架下添加其他条款，以满足 ccTLD 的特定需求。有关 DNSSEC 实践声明的示例，请参阅本指南附录 A。

3.3.3 区加密算法的选择

密码学领域在不断发展。当现有算法被发现不像以前认为的那样安全时，新算法将取代现有算法。因此，算法实施的要求和使用指南会定期更新，以反映新的现实。

实施 DNSSEC 需要选择适当的加密算法。在发布本指南时，RFC 8624，*Algorithm Implementation Requirements and Usage Guidance for DNSSEC*（*DNSSEC 的算法实现要求和使用指南*）提供了与 DNSSEC 相关的算法实现指南和签名参数要求。

下表列出了从 RFC 8624 中得到的一些（非详尽）建议。个别运营商可能有特定要求，可做相应调整。

项目	建议
DNSKEY 算法	**算法 13 (ECDSAP256SHA256)** 提供加密强度，是目前在新的 DNSSEC 部署中建议使用的算法，因为其密钥和签名较短，因此 DNS 数据包较小。 但**算法 8 (RSASHA256)** 也可使用，因为它被广泛部署，并且由于其加密强度多年来一直是默认算法。
授权签名者 (Delegation Signer, DS) 算法	SHA-256 被广泛使用，是一种强大的散列算法，因此建议用于 DS 的新部署和现有部署。
DNSSEC 安全算法（由加密算法和散列算法组成）	目前推荐的是算法 13 (ECDSAP256SHA256)。或者，也可使用算法 8 (RSASHA256)。
区签名密钥 (ZSK) 和密钥签名密钥 (KSK) 的密钥大小	算法 13 (ECDSAP256SHA256) 将始终生成 256 位的密钥。 算法 8 (RSASHA256) 的密钥大小可以设置在 2048 位至 4096 位之间。
ZSK 和 KSK 的有效期	由于运营商会根据其以往经验调整密钥有效期，因此没有很好的方法来预估不同运营商各自的需求。一些运营商会使用一个 ZSK 一到三个月，使用一个 KSK 一到五年，然后再轮转密钥。
私钥的存储	离线、未连接网络、物理安全的计算机，如硬件安全模块 (HSM)

注意：较大的密钥会增加 RRSIG 和 DNSKEY 记录的大小，因此也会增加 DNS UDP 数据包溢出的机会。此外，验证和创建资源记录数字签名 (RRSIG) 所需的时间随密钥的增大而增加，因此请避免不必要地增加密钥大小。

3.3.4 否定存在：NSEC 或 NSEC3

否定存在即证明某物不存在，是一种用于通知解析器某个域名不存在 (NXDOMAIN) 的机制。与之相反的情况是，域名存在，但它没有所请求的特定资源记录 (NODATA)。否定存在验证使用加

密对否定响应进行签名。在 DNSSEC 中，这可以分别使用 NSEC (Next Secure) 或 NSEC3 (Next Secure v3) 来实现。

NSEC 用于描述域名之间的间隔。它间接地告诉解析器哪些域名不存在于一个区，其方法是按规范顺序提供它之前的域名和之后的域名。以 NSEC 方式实施的这一机制构成了 DNSSEC 中否定存在验证的基础，但它面临两个问题：

- NSEC
  记录易受区遍历的影响，此弱点允许攻击者穿越区中的所有域名。因此，攻击者可以重建整个区，从而阻挠任何以管理方式阻止区传输的尝试。

- NSEC 面临的第二个问题是，在以授权为中心的区（如 TLD）中每个域名都会获得一条 NSEC 记录以及与之关联的
  RRSIG。对区进行签名之后，这将导致其大小的增加。这种增加产生的开销可能会对权威 DNS
  服务器的性能产生负面影响，例如限制硬件存储资源或延长执行区传输的持续时间。

NSEC3 则做出了改进，对域名进行散列处理并可利用加盐方式进一步强化，从而缓解了 NSEC 中的区遍历问题。此外，NSEC3 还提供一种称为“选择退出”的特定功能，可以使区中授权的未签名域名（不安全的授权）不需要 NSEC3 记录。这意味着，在 TLD 区上激活选择退出功能时，NSEC3 无法证明或否定在该 TLD 下注册的未签名域的存在。然而，NSEC3 的一个缺陷是 DNS 响应比 NSEC 的响应大。

在 NSEC 和 NSEC3 之间进行选择时，没有唯一正确的答案。如果为了防止区遍历而倾向于使用 NSEC3，则通常建议在没有额外迭代和空盐的情况下实现 NSEC3。但是，对于较小的区，不建议使用基于选择退出的 NSEC3 记录。对于非常大且稀疏的已签名区，如果其中大多数记录都是不安全的授权，则可以使用 NSEC3 选择退出功能。除了上述考虑事项之外，NSEC 比 NSEC3 更容易进行故障排除。

3.4 注册服务机构的参与

强烈建议在 ccTLD 签名的初步阶段就让注册服务机构参与进来。不仅因为他们是注册管理机构和注册人之间的中介，而且在 ccTLD 级别部署 DNSSEC 是保护该 ccTLD 域名空间的起点。对 ccTLD 进行签名后，二级域持有人就可以开始保护各自的域。这将要求注册服务机构能够从第二级或后续级别的域名持有人处收集一种新类型的记录，并发送给注册管理机构。这种新类型的记录称为“授权签名者 (Delegation Signer，DS)”。

从技术上讲，DS 是密钥签名密钥 (KSK) 的散列，有助于在 DNS 域名空间中的父区和子区之间建立信任链。在父区中拥有 DS 记录会创建一个安全链条，外部攻击者必须攻破该链条才能在子区中伪造密钥资料。

通常，ccTLD 与互联网号码分配机构 (Internet Assigned Numbers Authority, IANA) 共享其 DS 记录，以发布到根区。ccTLD 下的域名注册人和管理员则直接或通过注册管理机构与 ccTLD 共享其 DS 记录。注册服务机构在这里扮演两个重要角色：

- **提供一个安全可靠的界面或机制，从注册人处收集 DS 记录**；如果这种界面或机制尚不存在，有意愿向其注册人提供 DNSSEC 支持的注册服务机构应尽快实施这一界面或机制。没有标准化的方法在客户和注册服务机构之间移动 DS 记录。从简单的 Web 界面到各种 API，不同的注册服务机构有不同的机制。
- **将 DS 推送到注册管理机构。**建议采用自动化的解决方案将 DS 发布到父区，而不是手动干预。在注册管理机构-注册服务机构模型中，可以对可扩展供应协议 (EPP) 使用 DNSSEC 扩展来传输 DS 资源记录集 (RRset) 和可选的 DNSKEY RRset。无论任何情况，均应在 ccTLD 运营商和注册服务机构之间进行测试，以确保使用现有基础设施进行新交易是可行的。

另一种用于子区自动管理其与父区 DS 的机制是，如果父区对这些记录有接受政策，则使用 CDS（子 DS）或 CDNSKEY（子 DNSKEY）RRset。它们可用于以下三种用例：

- 初始 DS 发布
- 密钥轮转
- 回到不安全状态

简单地说，如果 CDS/CDNSKEY 和 DS 不同，则 CDS/CDNSKEY 是一条指示父区修改 DS 资源 RRset 的指令。RFC 8078，*Managing DS Records from the Parent via CDS/CDNSKey（通过 CDS/CDNSKey 管理源自父区的 DS 记录）*详细介绍了子区和父区之间 DS 记录的自动管理。

最后，让注册服务机构尽早参与这一过程也能让他们受益于 ICANN 及其互联网社群合作伙伴提供的 DNSSEC 培训和实践计划。

4 时间表

DNSSEC 部署没有具体的时间表。持续时间可能从几周到几个月或几年不等，具体取决于多个因素。但是，要避免长时间延滞，请考虑以下建议：

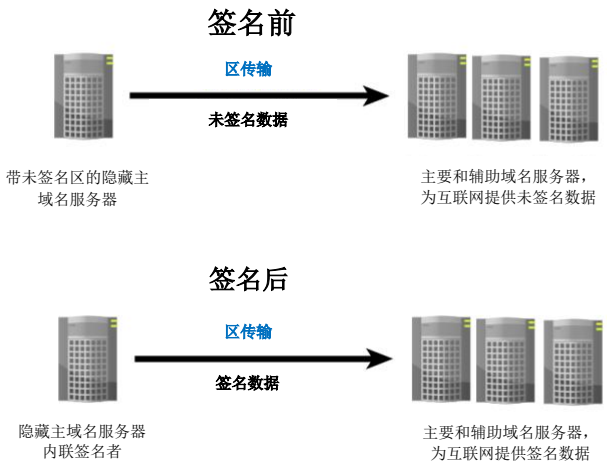
- 将这一过程作为一个项目进行定义，明确其开始日期、预计结束日期和要达到的里程碑。此外，还要为项目经理或技术负责人分配适当的资源
- 定义、管理并与利益相关方合作，包括监管机构、注册服务机构、技术和管理相关方、承包商、后端运营商或任何其他相关方。还需要解决不同 ccTLD 利益相关方之间的沟通问题
- 识别并妥善管理风险

5 DNSSEC 部署场景

无论您决定仅使用自己的资源和基础设施还是通过雇佣承包商（如后端运营商）来管理区，技术部署架构或许都可以仿效此处所述的两种主要场景之一。

5.1 使用主服务器（隐藏主服务器）内嵌签名

在此配置场景中，一个隐藏的主域名服务器通常不为互联网所知，为区提供一组 DNS 权威服务器，通常是一个公共主服务器和多个辅助服务器。隐藏的主域名服务器不是一项特定的 DNSSEC 要求，而是一种 DNS 运营最佳实践，建议使用带外权威域名服务器，该服务器对公众不可访问且未知，并且可以在其中执行该区的所有更新。此服务器还应实施更严格的安全和审计流程。架构类似下图所示：



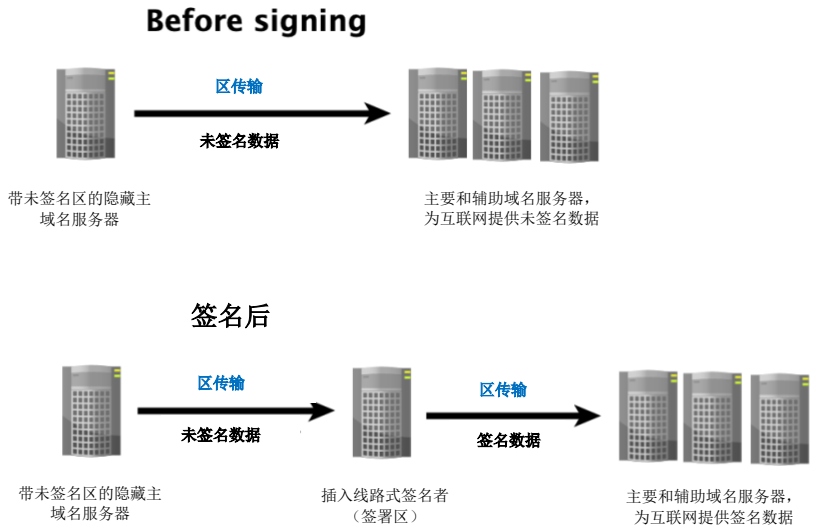
此隐藏的主服务器将被设置为识别创建的密钥，并按照其运行的 DNS 软件提供的流程，使用这些密钥来生成和运行签名区。完成后，签名版本的区文件将会被传输，并与所有适当的公共可见权威域名服务器保持同步。

除了隐藏的主域名服务器中的配置更改外，在此架构中没有其他要执行的配置或软件更改。

5.2 插入线路式 (Bump-in-the-wire, BITW) 内嵌签名

在此配置场景中，一个新的域名服务器（“签名者”）会插入到现有架构中，并放置在隐藏的主服务器与公共域名服务器之间（后者会将区提供给互联网）。这台新服务器充当一个插入线路式（bump-in-the-wire，BITW）设备。它从隐藏的主服务器获取未签名的区文件，对数据进行签名，然后将签名的区文件发送到域的公共域名服务器。

签名前



为此，可考虑以下步骤：

1. 隐藏的主服务器不应列在该域的域名服务器 (NS) 资源 RRset 中，以避免得到冲突的应答，即，来自隐藏主服务器的未签名应答和来自其他域名服务器的签名应答。
2. 应更新隐藏的主服务器配置，以便仅允许签名者服务器执行区传输。
3. 签名者服务器使用从隐藏的主服务器接收的未签名区文件和私钥对资源记录进行签名。最后，它使用区传输机制，将已签名的区文件分发到主域名服务器和辅助域名服务器。
4. 域名服务器的配置也应相应更新，以便仅从签名者服务器接收区传输。

无论部署场景如何，都建议在将签名区分发到所有服务器之前先对其进行验证。本文档第 10 节所述的在线工具可用来对域名进行验证。运行几次查询来搜索 DNSSEC 资源记录数字签名和签名到期日期，以及所管理的每个区上的其他参数，也是一种很好的做法。将运行这些测试作为验证您的实施的一部分。

最后，我们来比较一下普通 DNS 和 DNSSEC 在区生产链中的步骤：

- **普通 DNS**： 创建 → 验证 → 发布 → 监控
- **DNSSEC**： 创建 → 验证 → 签名 → 验证 → 发布 → 监控

6 对 TLD 进行签名

与进行其他重要更改时一样，我们强烈建议您在生产系统中规划部署之前，先进行深入的测试。实施 DNSSEC 时更是如此，因为新的更改会被引入到区中。下文只是简单概述了对 TLD 域进行签名过程中的常规步骤，实际根据您所处的环境和您的特殊要求，可能需要对这些步骤进行调整。

在某些情况下，一次性签署整个区文件不适用于大型 ccTLD；相反，制定一种递增式的方法来签署此区会更安全。其他 ccTLD 也可以选择先在测试平台上对一个子区进行签名，然后再将 DS 记录添加到父区中。这是实际签署 ccTLD 区之前的一项准备测试工作。

在任何情况下，要想成功完成签名过程，最好谨慎行事，采用合适的部署和测试计划以及严格的验证方法。

1. 部署和配置 DNSSEC 试验床环境。试验床可包含以下元素，具体视部署模型而定：
   - 一个测试签名服务器，和一个能够签署区文件的域名服务器（测试 NS）。该服务器要么能够自行生成签名密钥，要么能够接收从其他服务器或硬件安全模块 (Hardware Security Module, HSM) 生成的密钥，并且除了其他功能之外，还要能够对区进行签名并分发已签名的区文件。
   - 一个测试二级权威域名服务器，用于接收已签名的区并为其提供服务。
   - 应在本地配置一个测试解析器，为已签名的区执行 DNSSEC 验证。
2. 将未签名的区文件从隐藏的主服务器复制到测试签名服务器。在稍后的测试阶段，未签名的区文件可自动分发到测试签名系统。
3. 生成 KSK 和 ZSK 密钥。这两种密钥建议在签名系统之外生成。将 KSK 存储在 HSM 上或离线存储，并仅将其用于签名 DNSKEY 资源记录。
4. 对区进行签名并将其发布到测试二级域名服务器。
5. 生成 DS 并将其作为受信任的密钥导入到测试解析器中。在现实世界中，DS 不会直接由 TLD 管理员分发给世界各地的递归解析器；相反，它将被发送到 IANA 以供发布到根区。之后，世界各地的任何验证递归解析器都将从根区中获取与该 TLD 对应的 DS。
6. 根据定义的测试用例，通过执行用户验收测试 (User Acceptance Testing, UAT) 来运行一些测试。这些测试应涵盖诸如检索密钥和签名、检查签名到期时间、查询响应时间、大小等主题。执行 DNSSEC 验证及其他一些任务。
7. 如果测试签名的过程是自动化的，请观察签名的到期时间和新签名的自动生成情况。否则，您应该执行手动轮转，直到测试签名过程变成自动化过程。
8. 执行 ZSK 轮转和 KSK 轮转。轮转 KSK 时，生成新的 DS，然后通过将新的 DS 添加到测试解析器进行本地验证来模拟与父区的共享。根据测试环境中的密钥有效期等各种时间参数，您将需要从测试解析器中移除旧 DS，然后从区中移除旧 KSK 以完成 KSK 轮转。
9. 反复进行新的测试并调整和完善步骤 1 到 8。
10. 对测试方法感到满意之后，即可专注于正式启动和选择合适的环境与部署场景：BITW 或隐藏的主服务器内嵌签名。执行新的 UAT 以确认所有域权威域名服务器均能够正确地服务于该区及其对应的 DNSSEC 资源记录。
11. 最后，按照 IANA 在 https://www.iana.org/domains/root/manage 上发布的 TLD 授权管理指南，将 DS 发布到根区。一旦添加到根区，即向全世界宣布该 ccTLD 进行了 DNSSEC 签名，并且任何安全感知解析器都应对源自该区的 DNS 记录执行 DNSSEC 验证。强烈建议使用 https://dnsviz.net/ 之类的工具（详细信息请参阅第 10 节）来实际验证该区的 DNSSEC 信任链。
12. “正式签名”ccTLD 后，即可计划向注册服务机构开放访问权限，以便将注册人的 DS 记录发布到注册管理机构中，从而使信任链生效。

7 密钥与算法轮转

使用 DNSSEC 保护区时，无论是出于安全性或运营方面的考虑要对其进行定期替换，还是在紧急情况下进行替换，区经理人必须做好替换（即“轮转”）用于保护区的密钥的准备。要实施密

钥或算法轮转，需要引入新密钥并从区中丢弃旧密钥。轮转时非常重要的一点是，必须考虑存在于各种解析器缓存中的针对区发布的数据。忽略缓存中可能存在的数据有可能会导致客户端服务丢失。例如，考虑这样的情景：区数据是使用旧的密钥进行签名，但对该数据进行验证的解析器在自己的缓存中并没有这个旧的区密钥。如果旧密钥不再存在于当前区中，则验证将失败并且相应的区数据将被标记为伪造。

另一方面，如果解析器尝试验证使用新密钥签名的数据，而旧密钥仍然存在于该解析器缓存中，也会导致数据被标记为伪造。密钥和算法轮转技术有多种，例如 RFC 6781，*DNSSEC Operational Practices, Version 2（DNSSEC 运营实践第 2 版）*中所述的技术，以及 RFC 7583，*DNSSEC Key Rollover Timing Considerations（DNSSEC 密钥轮转时间安排注意事项）*中所述。此处简单列举其中几项技术，如：*pre-publish（预发布）*、*double-RRSIG ZSK rollover（双重 RRSIG ZSK 轮转）*、*double-KSK（双重 KSK）*、*double-DS（双重 DS）和 double-RRset（双重 RRset）*。

为应对特殊情况，例如因怀疑 ZSK 或 KSK 密钥被泄露而需紧急轮转密钥，建议先做好准备，事先制定相关的书面程序。

7.1 ZSK 轮转

在 ZSK 轮转期间，必须确保任何可以访问特定签名的缓存验证器也可以访问相应的有效 ZSK。RFC 6781“DNSSEC 运营实践第 2 版”中记录了执行 ZSK 轮转的三种方法：预发布、双重签名和双重 RRSIG。

在本文档中，我们只描述预发布方法，因为它在整个轮转过程中将区和响应的大小保持在最小。通过这种方法，新的 ZSK 被引入到 DNSKEY RRset 中，并且在经过足够长的一段时间之后，可以确保任何缓存的 DNSKEY RRset 都包含新旧两个密钥。之后，使用新的 ZSK 对区进行签名，并删除旧的签名。最后，当使用旧 ZSK 创建的所有签名都在缓存中过期时，旧密钥将被删除。以下步骤描述了这一过程。

1. 新的 ZSK A 被引入区并出现在 DNSKEY RRset 中，但尚未用来对该区中的记录进行签名。当前在用的 KSK 会弃用 DNSKEY RRset，并随后使用当前在用的 KSK 对 DNSKEY RRset 重新签名。在这一阶段，一般认为新的 ZSK 已经发布。
2. 一段时间后，ZSK A 即可对区中的记录进行签名。这里的“一段时间”，是区传播延迟的时间加上 DNSKEY 记录在该区中存在的时间。换句话说，它是现有 DNSKEY 记录从缓存中过期所需的最长时间。ZSK A 变为在用状态并生效，开始对该区的记录进行签名。
3. ZSK A 将继续签名和刷新该区的记录，直到需要发布新的 ZSK B 为止。密钥 B 的发布时间取决于 A 的激活时间及密钥管理政策中为该区设置的 ZSK 生命周期。ZSK B 已准备就绪，可用于对记录进行签名，但 ZSK A 仍处于在用状态。
4. 当密钥 A 到达 ZSK 生命周期时，即被淘汰。密钥 B 变为在用状态并用于对区进行签名。但是，过期密钥必须在区中保留一段时间（“退役缓冲期”），以允许使用该密钥生成的 RRSIG 可继续由解析器验证。退役缓冲期的时间长度为：所有现有的 RRset 使用密钥 B

重新签名所需的时间，加上区传播延迟的时间，以及该区中使用旧密钥创建的所有 RRSIG 的最大存活时间 (TTL)。
5. 一段时间后，使用过期密钥创建的签名会从解析器缓存中全部消失，此时认为旧密钥已彻底失效。
6. 旧密钥彻底失效后，即可从 DNSKEY RRset 中删除，DNSKEY RRset 将必须使用当前区的 KSK 重新签名。在此阶段，密钥 A 被宣告为已删除。
7. 一段时间后，新密钥将发布，并重复上述整个过程。

7.2 KSK 轮转

在 KSK 轮转中，主要挑战在于确保该区始终存在受信任的 KSK，即使在轮转过程中也是如此。RFC 6781“DNSSEC 运营实践第 2 版”中也记录了三种轮转 KSK 的方法：双重 KSK、双重 DS 和双重 RRset。双重 RRset 方法是其中最高效的方法，因为新的 DS 记录和 DNSKEY RRset 会并行传播。

通过这种方法，新的 DNSKEY 和 DS 记录可同时在相应的区中发布。当旧的 DNSKEY 和 DS RRset 从缓存中过期足够长的一段时间之后，它们就会从各自的区中删除。轮转步骤如下所述：

1. DS 和 DNSKEY 记录都存在于各自的区中。它们对应的 KSK A 处于在用状态，保护着该区的安全。
2. 一旦当前的 KSK A 生命周期临近，就会在区中引入一个新的 KSK B 并将其用于对 DNSKEY RRset 进行签名。KSK B 的 DS 被发送到父区中发布。
3. 父区可继续验证新的 DS，然后再将其发布到父区。
4. 经过一段时间后，新的 DS 或 DNSKEY 即已传播到验证解析器的缓存中。同时，ZSK A 将从区中删除。
5. 稍后，与 ZSK A 关联的 DS 和 DNSKEY 记录也会被删除。
6. 一段时间后，新密钥将发布，并重复上述整个过程。

8 签名区的其他注意事项

在准备 DNSSEC 部署策略时，需考虑以下要素：

- **制定能力培养计划**：确定并参与工作坊、网络研讨会、实践培训和任何其他有助于在 DNSSEC 运营方面增长知识和发展新技能的能力建设活动。ICANN 的技术合作提供此类培训，包括通常在 ICANN 会议期间举行的 DNSSEC 工作坊。
  - 除 ICANN 之外，网络新创企业资源中心 (Network Startup Resource Center, NSRC)、国际互联网协会 (Internet Society) 和地区互联网注册管理机构 (Regional Internet Registries, RIR) 也提供与 DNSSEC 相关的合作活动。
  - 最后，参加论坛、网络研讨会和工作坊，在这些活动中，无论经验丰富的运营商还是新成员都会出席，大家可当面探讨当前和今后的 DNSSEC 部署，从而大大增加您对 DNSSEC 运营实践的了解。
- **订阅网络运营商团体 (NOG) 电子邮件清单**：这些都是很好的论坛，大家在这里讨论和分享技术经验和专业知识，并在技术问题上寻求支持和帮助。因此，可以将其视为一个在需要时可为您提供帮助的社群。

- **密钥生成和管理**：硬件安全模块 (Hardware Security Modules, HSM) 往往可以为生成和存储密钥提供良好的设施。然而，购买、保护和维护 HSM 的成本值得考虑。视具体功能而定，HSM 的成本可能从数百美元到数千美元不等。HSM 可能还会增加培训开销，因为学习任何新硬件都具有一定的挑战性。使用 HSM 是一种很好的做法，但不是生成密钥的唯一方法。另一种值得考虑的可能性是在离线、无网络连接且物理安全的计算机中生成、存储和使用密钥。RFC 6781 *DNSSEC Operational Practices, Version 2"DNSSEC 运营实践第 2 版（）*中提供了有关密钥生成和管理的更多详细信息。
- **时间安排注意事项**：DNSSEC 在 DNS 中引入了绝对时间的概念。DNSSEC 中签名的有效期从签名生成之日起，到签名到期之日止，此后签名将被标记为无效且签名数据被视为伪造数据。请务必妥善管理时间，这样才能在正确的有效期内生成签名。想象一下，如果签名区的签名已过有效期，这将导致解析器端的验证失败。因此，强烈建议配置网络时间协议 (Network Time Protocol, NTP) 服务器以保持准确的时间。还要注意其他事项，例如 RFC 6781 *DNSSEC Operational Practices, Version 2（"DNSSEC 运营实践第 2 版）*中描述的最小和最大区存活时间 (TTL) 及签名发布期和签名有效期。
- **软件、硬件和网络要求**：目前，开源的和商业的 DNSSEC 实施均受支持，包括伯克利互联网域名 (Berkeley Internet Name Domain, BIND)、PowerDNS、NLnet Labs Name Server Daemon (Name Server Daemon, NSD) 和 Knot DNS 等。OpenDNSSEC 是一种仍被广泛使用的签名解决方案，因为它可以自动化跟踪 DNSSEC 密钥和区签名的过程。如果您计划在权威服务器上部署 DNSSEC，则需要在系统上生成加密签名密钥。生成密钥所需时间长短取决于系统中的随机性（熵）来源。熵不足的虚拟机等系统可能需要更长的时间来生成密钥。
  - CPU、系统存储和内存等硬件资源也值得考虑是否要进行优化。这是因为启用 DNSSEC 会增加系统存储、内存占用和 CPU 负载，部分原因在于密钥的生成和签署。签名区的区文件大小总是会大幅增长。
  - 在网络安全防护策略方面，请验证防火墙和访问控制列表 (Access Control Lists, ACL) 规则，例如，在端口 53 上允许大型 DNS UDP 数据包和基于 TCP 的 DNS。此外，需要分别在 DNS 服务器和网络配置中激活 DNS 扩展机制 (Extension Mechanism for DNS, EDNS0)。

9 必要时取消对 TLD 的签名

与世界上的许多事物一样，DNSSEC 也并非完美无瑕。DNSSEC 增加了 DNS 的复杂性，因此也增加了出现问题或出错的可能性。例如，KSK 或 ZSK 甚至两者都可能丢失或被盗用。意外的硬件或软件错误可能会阻止区被签名和分发，从而影响区获得正常服务。

在最坏的情况下，您可能更愿意取消对区的签名，以便在再次签名之前，修复所有的问题和错误。但是，取消区签名会由于将其恢复到不安全状态而产生成本。

全世界都知道，某个区是否通过父区中存在的 DS 记录进行了签名或取消签名。如果 DS 记录不存在，则信任链就没有保障。因此，恢复到未签名状态从技术上讲很简单，就是从父区中删除所有 DS 记录。对于 ccTLD，这意味着请求 IANA 从根区中删除相应的 DS 记录。

解决所有问题后，TLD 运营商应考虑生成新密钥并重新对区进行签名。此外，在将新的 DS 记录发布到根区之前，运营商应确保新签名的区在所有域名服务器中分发良好且可用。IANA 发布 DS 记录后，整个互联网就会知道该 TLD 区已重新签名，并且任何验证解析器都将验证该区提供的所有资源记录。

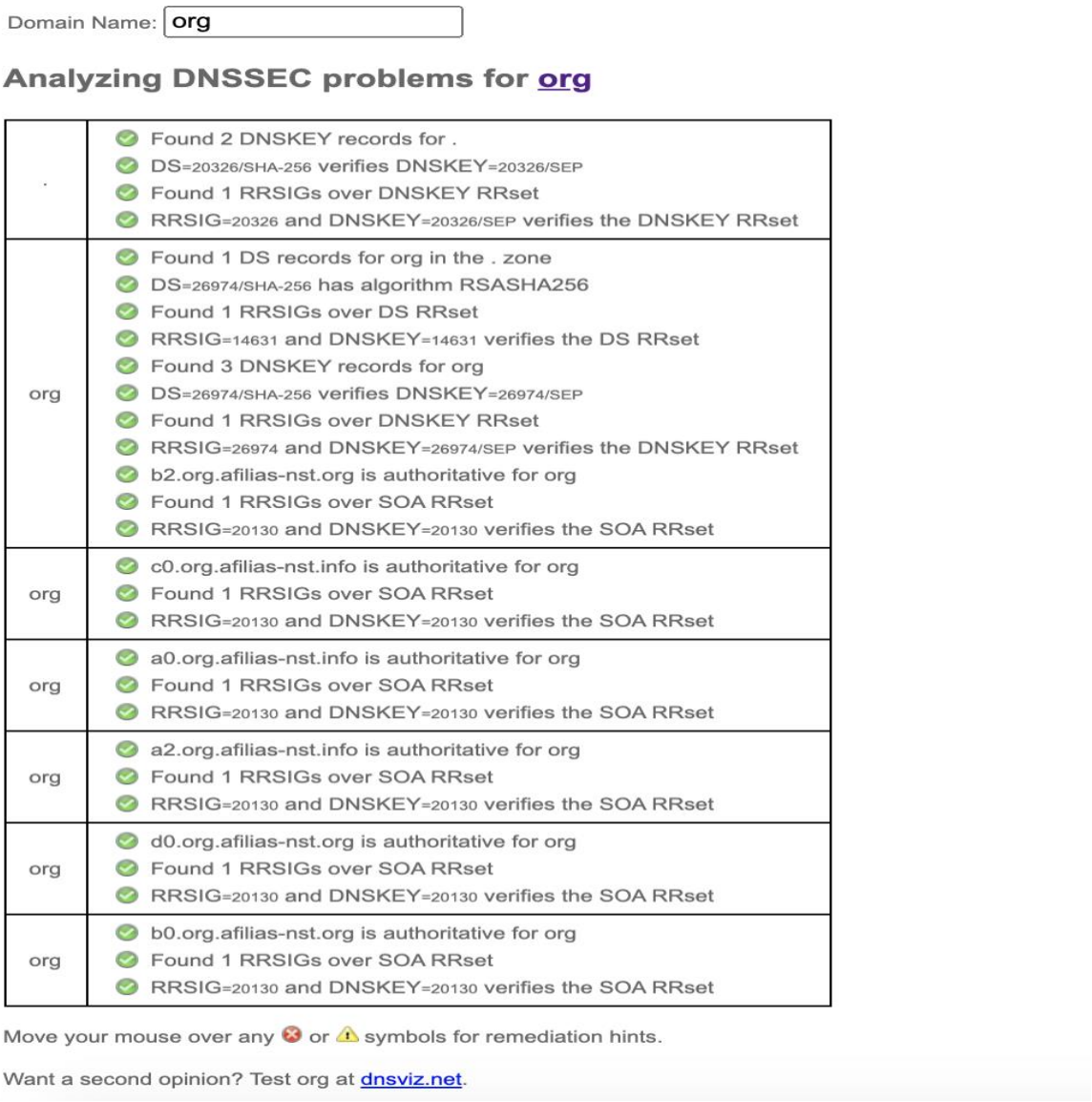
10 有用的 DNSSEC 工具

以下工具可用于对 DNSSEC 问题进行故障排除。

10.1 Verisign DNSSEC 调试器

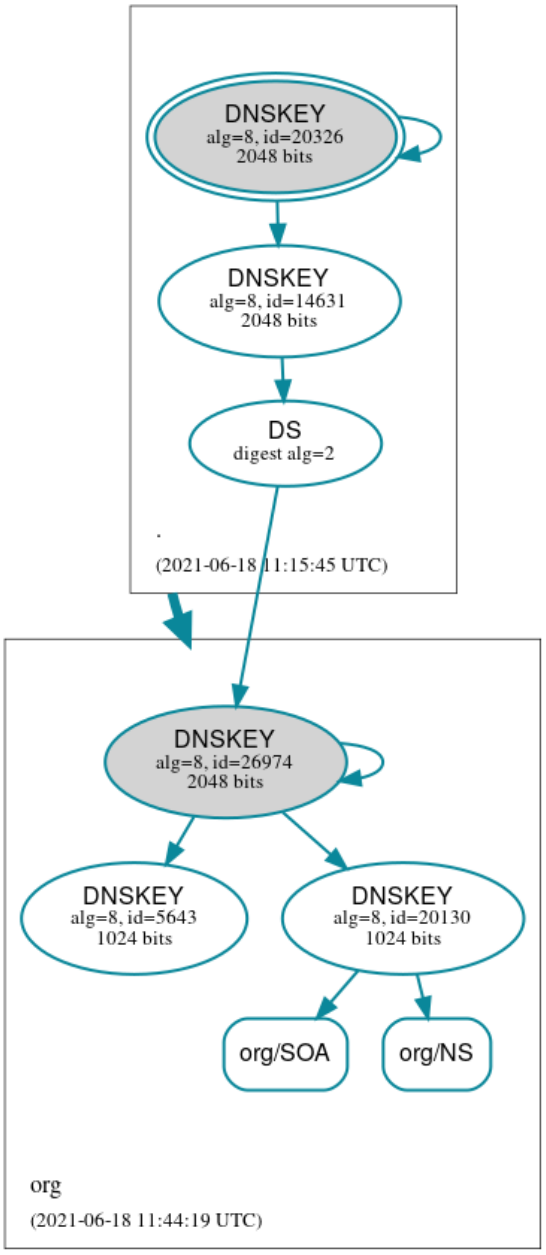
这款 DNSSEC 调试器是一个基于 Web 的工具，有助于确认某个具体的已启用 DNSSEC 的域名其 *“信任链”* 是否完好无损，工具地址为 https://dnssec-debugger.verisignlabs.com/。它可以对给定的域名进行逐步验证并突出显示任何发现的问题。

下面是一个输出示例：



10.2 DNSVIZ

DNSViz (https://dnsviz.net) 是一个可视化呈现 DNS 区状态的工具。它可以针对某个域名的 DNSSEC 认证链，以及该域名在 DNS 域名空间中的解析路径，提供可视化分析。该工具检测到的配置错误也会一并列出。以下是 .ORG 区的可视化图表：



11 结语

作为一个强大的协议，DNSSEC 可以为 DNS 数据提供来源认证和数据完整性。DNSSEC 引入了普通 DNS 中不存在的新概念和新任务，因此，使用 DNSSEC 对域名进行签名会在运营层面上带来很多变化。

本文档作为一份指导手册，可帮助 ccTLD 注册管理运行机构和任何其他相关方了解对 ccTLD 进行签名的相关概念和步骤。关于 DNSSEC 有很多可以讨论的内容，本文档仅涵盖一些有关部署的最重要方面。

与其他安全性解决方案一样，我们建议您按照正确的流程进行操作并做好准备，以避免出现问题。各方都必须评估其自身环境及相关的威胁和漏洞，以确定他们在依靠 DNSSEC 保护区和其中的域名时，愿意承受的风险级别。

从本质上讲，所有利益相关方的协调努力与积极合作仍然是成功部署 DNSSEC 的关键。

A：DNSSEC 政策和 DNSSEC 实践声明示例

- ZACR DNSSEC 政策和实践声明框架，第 001 版，2016 年 9 月，南非中央注册管理机构 (ZACR)：https://www.registry.net.za/downloads/u/zacr-dps-signed.pdf
- .fr DNSSEC 实践声明，第 1.2 版，2013 年 6 月，法国互联网域名与合作协会 (Afnic)：https://www.afnic.fr/wp-media/uploads/2020/12/dps-english-fr.pdf
- .CA CIRA DNSSEC 实践声明，第 1.5 版，2016 年 8 月，加拿大互联网注册管理局 (CIRA)：https://www.cira.ca/cira-dnssec-practice-statement-ca
- JP 区 DNSSEC 实践声明 (JP DPS)，第 1.4 版，2015 年 10 月，日本注册管理机构服务公司 (JPRS)：https://jprs.jp/doc/dnssec/jp-dps-eng.v1.4.html
- 适用于 TLD/GTLD 区的 Verisign DNSSEC 实践声明，第 1.8 版，2019 年12 月，威瑞信公司 (Verisign Inc.)：https://www.verisign.com/assets/20191111_CTLD_VerisignDNSSECPracticeStatement_v1.8_finalized.pdf。这些区的列表可在以下位置找到：https://www.verisign.com/assets/20190430-Verisign-Operated-TLD-GTLD-Zones-v1.04-Converted.pdf

B：未签名区和已签名区示例

B.1 未签名区

```
example.                  86400  IN     SOA    a.nic.dns.blablabla.
hostmaster.dns.blablabla. (
                2017072300 ; serial
                1800       ; refresh (30 minutes)
                900        ; retry (15 minutes)
                2419200    ; expire (4 weeks)
                300        ; minimum (5 minutes)
                )

example.                  86400  IN     NS     a.nic.dns.blablabla.
example.                  86400  IN     NS     b.nic.dns.blablabla.
example.                  86400  IN     NS     d.nic.dns.blablabla.
example.                  86400  IN     NS     e.nic.dns.blablabla.
example.                  86400  IN     NS     f.nic.dns.blablabla.
```

```
aaa.example.              86400  IN     NS     ns1.reg.zzzz.
aaa.example.              86400  IN     NS     ns2.reg.zzzz.
bbb.example.              86400  IN     NS     ns1.reg.zzzz.
bbb.example.              86400  IN     NS     ns2.reg.zzzz.
ccc.example.              86400  IN     NS     ns1.reg.zzzz.
ccc.example.              86400  IN     NS     ns2.reg.zzzz.
ddd.example.              86400  IN     NS     ns3-l2.nic.zzzz.
```

B.2 已签名区

```
example.                  86400  IN     SOA    a.nic.dns.blablabla.
hostmaster.dns.blablabla. (
                2017072305 ; serial
                1800       ; refresh (30 minutes)
                900        ; retry (15 minutes)
                2419200    ; expire (4 weeks)
                300        ; minimum (5 minutes)
                )

example.                  86400  IN     RRSIG  SOA 8 1 86400 20170724231821
20170618015310 660 example. nQ8H8StSRDoQgzwBNQ0k9+E1LGrV0tsCinoB6KxcyuHfGT4ehWsj5JI6
N01WpXqy/q1S/XlhqtjVoiti4zSOwIjFlSloug3W09eJnH9biwmb6U8B
JQoHf3edGvZtWNZdtcOKY1CFBI2ApceFn8KOYvT0qzpygOlF5lMrJvnO J5c=
example.                  86400  IN     NS     a.nic.dns.blablabla.
example.                  86400  IN     NS     b.nic.dns.blablabla.
example.                  86400  IN     NS     d.nic.dns.blablabla.
example.                  86400  IN     NS     e.nic.dns.blablabla.
example.                  86400  IN     NS     f.nic.dns.blablabla.
example.                  86400  IN     RRSIG  NS 8 1 86400 20170730192856
20170617025305 660 example. KNaF2jTPuCGq5FIzspbJL+TDBx/6z01E7+tkkzYRNh0xAKDnutcfb1It
D7XrNWPEbXsaafFyZ/M5DaDGzTzsVNm1h9h3md6o0vZNH07q8nmm+fYX
do8sx9aFxCgl9NsmG0cyrBbVnyrPKxDlAx69HJCh0kBb7PFKhr1hpnYY xGA=
example.                  86400  IN     DNSKEY 256 3 8
AwEAAasHjitdDurpevNLojD4Sp3609P+C9uOTR42DJel0NSSva/x38Ba
7gs0b4Q+tmKPI5cmxDhECiUfdzaRRA8vxPZK8x5LL/V1WZ5q6egFmH4x
eLxWaxlftFotev/T8kVe7jZUk7Hh3x7LPgGLajpjNNFELj42Xe6XBkkN 9FY1lQkB
example.                  86400  IN     DNSKEY 257 3 8
AwEAAY6HLDY5M5kjlrjVV9HQyWUkkrYZ2eB8KeJjUMN9qDM6FsA57pbS
5tmbGV1zxxqGon0pO7HYV06GZIgFOLBqDvgGsnKDQ5A2iktYNuSmTh+w
fd8ixgbYigtoBMBnNeqFozMK58c1yf7amui2cCOg9ibGZMpLQvjKOSyV
Jnlh0l8e3OE7Ul1GEa39XpVez2wkjImhsG0e7KAZPlFjEUpvwie8HEQV
jz3PK7Zr6SZVLLyet0rnN3prCHfvhNh6DycN/rt6/PopLvPQM8SaW+u8
zn6Z4S4AoTPTxKm5udzb7mWf7lT83PAbOvLu/WIRY6nqye+4SkJsrmjI xnLdk/Q54E8=
example.                  86400  IN     RRSIG  DNSKEY 8 1 86400 20170703000000
20170613000000 54322 example. B2riGYos+/q5RqXVBQKrrkVUuruDBH8ANNa8J6sMHUjf0MPZOuICd2kZ
PlAGMpZpp8LoaRoG2zaTVILZ8Vhi90FsyLsZVpPooAvmKlTFOrWoJoPo
XScLhb3ISRLOzKEnyLt5Ds3TxuabHLPlf8jpTXaHMFZCzYYtTJJQb+M3
BLeK+Lx4uCwU1pvxNkuR9StKa5tJquByIZCWZsSx5nKWPyrsGLtFJKrg
DXe8XlA8LxeER69OQgSZ1VXvK8Kd4p3wyvzUHCcsPYZzebxHXPqDrYB7
BU7eqsDUjCfThqbkC0Ju7koHROYRjGdoY/4f6nDOJEOICIFEgEDHJg2t w1nENQ==
example.                  0      IN     NSEC3PARAM 1 0 3 00FF
example.                  0      IN     RRSIG  NSEC3PARAM 8 1 0 20170716213640
20170606005304 660 example. a6Mp1NjW2/nnn+5i98AWzVrOX0yUvu/urP1cqY6zZjISReZOSLx6aorJ
lM9Nnx1fNVr2COtD71UVJI7kFUC5jVbmAitWdHHH/zyzK6WyyaN5Nsaf
cKW0Su8lLkctCHiqpKmuhOhnK1Dqmigx8YhyhPbN5nCzoST6lcnNjtV0 TwQ=
```

```
aaa.example.        86400  IN     NS     ns1.reg.zzzz.
aaa.example.        86400  IN     NS     ns2.reg.zzzz.
bbb.example.        86400  IN     NS     ns1.reg.zzzz.
bbb.example.        86400  IN     NS     ns2.reg.zzzz.
ccc.example.        86400  IN     NS     ns1.reg.zzzz.
ccc.example.        86400  IN     NS     ns2.reg.zzzz.
ddd.example.        86400  IN     NS     ns3-l2.nic.zzzz.
0KPQJ71AL5RHRST9HM8LEFLK0I0QN5N7.example.      3600 IN      NSEC3 1 1 3 00FF
464L7A368JEOCPKU9G34B9RQADEPKA14 NS DS RRSIG
0KPQJ71AL5RHRST9HM8LEFLK0I0QN5N7.example.      3600 IN      RRSIG NSEC3 8 2 3600
20170703210235 20170602012306 660 example.
H+qdaHqnAgUa66VSKmMmfKWopeZQM0ridMUN2YN4rncHeWD8b0yA6O6N
hLF/ojpZoGrQN+G+p4SWJVb/pj2CkLk00E2AhloXXV0KaQIzUwPVNm7p
J9es7ohi5ErGtM1ClLpGggz05qNWboejbrXtS8TFdoTtn6Z2Omk4RNmj hG0=
464L7A368JEOCPKU9G34B9RQADEPKA14.example.      3600 IN      NSEC3 1 1 3 00FF
MLTMB5J4Q7T5R3GJBSBTMVD2LBMFU3KA NS DS RRSIG
464L7A368JEOCPKU9G34B9RQADEPKA14.example.      3600 IN      RRSIG NSEC3 8 2 3600
20170715005821 20170610062309 660 example.
dk6WScB3zmJYig0w8LxFXoc9vj1leqFRBlET4YAVVmeAwcGf0ixa41T+
pKKcMHbXDsw+PHYZHaRLmA9lEgs+4lJMdA3fRroNSXyV2usHMdFaKUoG
UZKehVGdgrBRx4vx+o4w1ztdumY6MsD0ART6IrhUbr+cvGHAlxNSviCI BbE=
MLTMB5J4Q7T5R3GJBSBTMVD2LBMFU3KA.example.      3600 IN      NSEC3 1 1 3 00FF
0KPQJ71AL5RHRST9HM8LEFLK0I0QN5N7 NS SOA RRSIG DNSKEY NSEC3PARAM
MLTMB5J4Q7T5R3GJBSBTMVD2LBMFU3KA.example.      3600 IN      RRSIG NSEC3 8 2 3600
20170706043605 20170604225320 660 example.
Ndq6p+Y8ztlgNN1vH12o5rxxh7QM8GLY3E1FPCX4h7N4RtnuoPpvEpsl
/K4XQ1p/8Uehe6Izg0BpvQ7A256/+UW3lkwlonR7UaOX/+gkEdxuxlC/
41nX5fI9G5QFrV7H8B7ezlVF/uLz4nXyH4mzz496x4iTMEoHfoAdMinL C7A=
example.                  86400  IN     SOA    a.nic.dns.blablabla.
hostmaster.dns.blablabla. 2017072305 86400 14400 2592000 3600
```

C：DNSSEC 部署工作清单

C.1 启动和准备

- ❑ 将 DNSSEC 部署作为一个项目进行定义

  运用项目管理方法，将整个部署过程作为一个有开始日期、预计结束日期和明确可交付成果的项目进行管理

- ❑ 记录现有系统。

  通过一份及时更新的书面文档，对系统和正在使用的流程进行描述

- ❑ 审核现有基础设施

  在 DNSSEC 部署之前修复当前系统中的任何缺陷

- ❑ 让利益相关方参与其中

  让他们充分了解 DNSSEC 部署并为部署做好准备

- ❑ 制定并遵循培训计划

  为您的员工和利益相关方提供 DNSSEC 部署所需的知识和技能

C.2 部署和监控

- ❑ 编写 DP 或 DPS

  发布适用于区的 DNSSEC 运营框架
- ❑ 编写 DNSSEC 部署计划

  书面记录全局部署策略和步骤。此文档可解决部署清单中的几个项目。
- ❑ 编写并验证 DNSSEC 流程和运营程序

  针对您的具体要求和环境，书面记录相关程序
- ❑ 选择 DNSSEC 部署场景

  确定 DNSSEC 实施模型
- ❑ 确定并购买新材料和设备

  获取新设备和材料
- ❑ 选择 DNSSEC 签名参数

  为一组 DNSSEC 参数分配值
- ❑ 搭建 DNSSEC 测试平台

  编写、运行和验证关于区签名的测试用例以熟悉 DNSSEC 流程和运作
- ❑ 计划正式启动并做好回退准备

  做好在生产环境中正式启动、监控和回退的准备，并就相关程序做好文档说明
- ❑ 正式启动和监控

  在生产环境中实施并监控 DNSSEC 签名
- ❑ 计划注册人的 DS 发布

  推广 DNSSEC，准备接收和处理子域名 DS 记录
- ❑ 发布注册人 DS

  公告 ccTLD 区中的子域 DS
- ❑ 记录经验教训

  总结和整理此过程中每一步的经验和教训

D：拓展阅读

- ⊙ *重大 DNSSEC 中断与验证失败事件*，IANIX：https://ianix.com/pub/dnssec-outages.html
- ⊙ *DNSSEC：算法部署道阻且长*，APNIC：https://blog.apnic.net/2020/12/01/dnssec-the-long-and-bumpy-road-of-algorithm-deployment/
- ⊙ *DNSSEC 基础设施审核框架*，NLnet Labs：https://nlnetlabs.nl/downloads/publications/dns-audit-framework-1.0.pdf

- *根 DNSSEC 信息*，IANA：https://www.iana.org/dnssec
- *DNSSEC 常见问题解答*，SIDN：https://www.sidn.nl/en/faq/dnssec
- RFC 6781，*DNSSEC Operational Practices v2（DNSSEC 运营实践第 2 版）*：https://www.rfc-editor.org/rfc/rfc6781.html

1. RFC 6841，*A Framework for DNSSEC Policies and DNSSEC Practice Statements（用于 DNSSEC 政策和 DNSSEC 实践声明的框架）*：https://www.rfc-editor.org/rfc/rfc6841.html

2. RFC 8078，*Managing DS Records from the Parent via CDS/CDNSKey（通过 CDS/CDNSKey 管理来自父级的 DS 记录）*：https://www.rfc-editor.org/rfc/rfc8078.html
3. RFC 8624，*Algorithm Implementation Requirements and Usage Guidance for DNSSEC（DNSSEC 的算法实现要求和使用指南）*：https://www.rfc-editor.org/rfc/rfc8624.html