BlindIDS：
加密流量上符合市场要求和隐私友好的入侵检测系统

塞巴斯蒂安·卡纳德（Sebastien Canard）
法国
sebastien.canard@orange.com

Aida Diop
橙色实验室法国橙色实验室
aida.diop@orange.com

Nizar Kheir
Thales Group，法国
nizar.kheir@thalesgroup.com

玛丽·潘达德文（Marie Paindavoine）

穆罕默德·萨伯特（Mohamed Sabt）

抽象的

网络入侵检测的目的是检查网络流量，以识别威胁和已知的攻击模式。它的主要功能之一是深度数据包检测（DPI），它可以提取网络数据包的内容，并将其与一组检测签名进行比较。尽管 DPI 通常用于保护网络和信息系统，但它需要直接访问流量内容，这使其对诸如 HTTPS 之类的加密网络协议视而不见。到目前为止，要通过检查网络用户的流量内容以检测攻击或恶意活动，在网络用户的隐私和安全性之间做出艰难的选择。

本文提出了一种新颖的方法，可以弥合网络安全和隐私之间的差距。它使直接对加密流量执行 DPI 成为可能，而无需知道流量内容或检测签名的模式。我们工作的相关性在于，它保留了安全市场生态系统中的微妙平衡。实际上，安全编辑器将能够保护其独特的检测签名，并仅通过加密的攻击模式来提供服务提供商。此外，服务提供商将能够在其体系结构中集成加密签名并执行 DPI，而不会损害网络通信的隐私性。最后，用户将能够通过流量加密来保护自己的隐私，同时还能从网络安全服务中受益。

AsiaCCS '17，2017 年 4 月 2 日至 6 日，阿拉伯联合酋长国阿布扎比

ISBN 978-1-4503-4944-4 / 17/04 ... 15.00 美元

DOI：http：//dx.doi.org/10.1145/3052973.3053013

关键词

入侵检测，网络中间盒，深度包检查，可解密的可搜索加密，安全性，隐私

1. 引言

如今，网络入侵检测和防御系统已广泛用于 IT 系统的早期预警和保护，以防一般攻击和入侵。它们可以识别系统内部人员和外部攻击者对信息系统的未经授权的使用，滥用和滥用[22]。它们还提供了广泛的功能，例如通过阻止未经授权的内容和目的地（例如 IP 或域黑名单）进行访问控制，数据丢失保护以及检测网络攻击（例如漏洞利用，恶意软件和僵尸网络）。这些系统的主要功能是深度包检查（DPI）。它根据与一组已知恶意模式的细粒度匹配，检查网络数据包的内容并调整决策，包括发出警报甚至拒绝数据包。如今，安全编辑器操作广泛的威胁情报，以构建和更新其全面的攻击特征集，然后进一步用于执行 DPI 和提供增值安全服务。实际上，大多数安全编辑者提出了他们的攻击特征数据库作为关键的竞争优势，认为它们可能涵盖了更广泛的恶意模式和攻击[13，5]。实际上，它们通常基于攻击签名的覆盖范围而彼此进行基准测试，这构成了基于 DPI 的安全服务的关键驱动力-例如[26]。大多数安全编辑者提出了他们的攻击特征库，作为关键的竞争优势，他们认为它们可能涵盖了更广泛的恶意模式和攻击[13，5]。实际上，它们确实经常根据攻击特征的覆盖范围进行基准测试，这构成了基于 DPI 的安全服务的关键驱动力，例如[26]。大多数安全编辑者提出了他们的攻击特征库，作为关键的竞争优势，他们认为它们可能涵盖了更广泛的恶意模式和攻击[13，5]。实际上，它们通常基于攻击签名的覆盖范围而彼此进行基准测试，这构成了基于 DPI 的安全服务的关键驱动力-例如[26]。

但是，DPI 需要访问网络数据包的明文内容，从而使其对网络加密（例如，使用 SSL / TLS 协议）视而不见。这是 DPI 的基本限制，因为它为大量攻击打开了大门。根据最近的一项研究，在使用加密作为掩盖的同时，2015 年将近一半的恶意软件攻击已渗透到目标网络[27]。这是在网络加密趋势日益发展的背景下进行的，到 2016 年底，全球 Internet 流量的 70％被加密，而某些网络超过 80％[24]。

为了克服网络加密的挑战，一种称为 SSL 检查的通用解决方案提供了在网络连接的两端之间使用中间人（MITM）设备的功能。它使用受信任的证书来模拟原始 SSL 会话的接收者，然后在明文流量限制中解密并应用 DPI。

帐篷[14]。尽管如此，SSL 检查破坏了 SSL 协议的端到端安全性，从而引发了有关 MITM 设备生成的日志数据的保密性的安全性问题[15]。这也引发了道德问题。例如，谷歌发现了未经授权使用由与法国情报机构合作的法国网络安全机构 ANSSI 链接的中间证书颁发机构颁发的数字证书[23]。

到目前为止，要在安全性和隐私之间做出微妙的选择。保留端到端加密，或使用 SSL 检查和 DPI [16]。最近，一种称为 BlindBox 的新方法提出了一种加密协议，该协议首次使直接在加密流量上搜索恶意模式成为可能[25]。BlindBox 基于多方计算技术，例如电路乱码和遗忘传输。它允许加密连接两端之间的网络路径上的中间盒设备搜索恶意模式，而无需破坏端到端加密。这是 BlindBox 的主要贡献，因为它在协调网络安全性和隐私性的漫长道路上树立了里程碑。尽管有原始解决方案，

首先，BlindBox 要求中间盒使用从每个新 HTTPS 连接的秘密会话密钥派生的密钥对要在流量中寻找的整个恶意模式进行加密。这大大增加了连接建立的时间。尤其是，在使用 3000 个检测签名的样本集的情况下，对于通用 HTTPS 连接，连接建立时间在[25]到 97 秒之间进行了评估。在每个 HTTPS 连接的整个持续时间内，中间盒还需要加密和管理用于检测的整个恶意模式集的不同副本。这在中间盒上需要过多的存储空间（例如，用于 3K 规则的 512GB 和 100 个并行连接），这使其很难集成到任何实际部署中。第二，BlindBox 要求安全编辑者以明文形式将其整个恶意模式集提供给负责中间盒设备的服务提供商。尽管从技术角度来看这没有什么区别，但它不符合安全市场生态系统中的微妙平衡。这种恶意模式及其覆盖范围构成了任何安全编辑人员的关键业务区分因素。因此，安全编辑人员极不可能热衷于将这些有价值的资产以明文形式传递给中间盒设备[19]。这种恶意模式及其覆盖范围构成了任何安全编辑人员的关键业务区分因素。因此，安全编辑人员极不可能热衷于将这些有价值的资产以明文形式传递给中间盒设备[19]。

在本文中，我们利用基于公钥密码学的可解密可搜索加密方案[12]，解决了 BlindBox 方法的这些主要局限性。主要目标是提供一种技术上合理的解决方案，该解决方案同时（a）对隐私友好，这意味着无法访问加密流量的明文内容，（b）具有安全意识，这意味着它支持 DPI over 加密流量，以及（c）实用，可以同时满足性能和实际市场需求。本文进行的广泛评估表明，我们的加密协议将 BlindBox 的性能提高了几个数量级，包括连接建立时间和中间盒执行其任务所需的内存空间。

总而言之，本文的贡献有三点：

•同时实现隐私和安全的 DPI 解决方案

严格性要求，与其他类似的加密方案（如 BlindBox）相比，性能更接近实际条件几个数量级；

- DPI 解决方案，可同时保留网络流量和特征码签名的私密性。这确实为用户提供了更多的隐私保证，因为中间盒甚至不知道其在流量中寻找的模式。
- 第一个完整而正式的安全模型，该模型描述了在加密流量上运行的入侵检测系统的特征，这种贡献可能与个人利益无关。我们还证明了我们的 DPI 解决方案在此模型中是安全的。

现在该文件的组织方式如下。第 2 节介绍了我们解决方案的体系结构以及旨在实现的关键安全性和隐私要求。第 3 节回顾了相关工作并总结了我们的主要贡献。第 4 节介绍了我们的加密协议，并提供了适当的安全性和隐私证明。第 5 节介绍了评估和实验结果。最后，第 6 节总结。

2. 建筑与安全

我们首先描述将要使用的主要体系结构，然后给出在加密网络通信的情况下入侵检测系统应满足的必需安全属性。

2.1 全球架构

我们在本文中描述的加密协议涉及四个主要角色：安全编辑器（在[25]中称为规则生成器），服务提供者（在[25]中又称为中间盒），发送者和接收者。就我们的解决方案而言，发送者和接收者角色可以互换安全网络连接的每一端都可以在同一会话中扮演发送方和接收方的角色。在标准入侵检测方案中，发送者或接收者角色可能是试图破坏另一个远程和良性实体的恶意实体（例如，单个攻击者，受感染的服务器）。其他场景，例如包括连接到恶意目的地的被恶意软件感染的终端，仍然可能是可行的。在这种情况下，发送者和接收者角色都是恶意的。除非两个实体能够一起协作，否则这也是我们在加密协议中涉及的场景。尽管从技术上讲是可行的，但这种情况不能视为对我们方法的限制，因为双方都可以通过安全编辑者和服务提供者都无法达到的第三种渠道就秘密密钥达成一致。因此，即使在明文网络流量的情况下，此类攻击也不会引起注意。

安全编辑器是负责编辑检测签名的实体。这些签名可能包括 IP 或域黑名单，例如恶意软件域，注册到僵尸网络中的 IP 或传递受审查或非法内容（例如通奸和恐怖主义）的网站。它们还可能包含二进制模式的逻辑组合，以捕获恶意软件样本和漏洞利用程序。最后，它们可能包含更精细的正则表达式，这些正则表达式表征了诸如 SQL 和代码注入之类的细粒度攻击。检测签名是安全性编辑器的关键资产。

他们需要广泛而持续的威胁情报，以便不断扩大覆盖范围，因此它们可能会捕获包括零时差攻击在内的新威胁。本文的安全编辑者角色可能属于众多市场竞争对手，例如 Symantec，趋势科技，McAfee，BitDefender 和 Kaspersky。

另一方面，服务提供商是为最终用户和企业提供网络和安全服务的利益相关者。它们提供中间盒，既是物理设备又是基于虚拟云的服务，并且支持多种安全服务，例如防火墙，代理，数据丢失保护，入侵检测和防御。服务提供商通常与安全编辑器合作，以便将检测逻辑（例如模式签名）添加到其中间盒中。在本文的上下文中，要求使用深度数据包检查的服务为加密和明文网络连接提供相同的安全保证。

2.2 安全要求

在进行更正式的描述之前，我们首先对预期的安全要求说几句话。

服务提供者（SP）角色在我们的协议中作为*诚实但好奇的*实体处理。它使用安全编辑器提供的检测签名，将 DPI 诚实地应用于加密流量。该 SP 会尝试，但是，以获取有关信息，一方或双方的业务内容;以及在签名代表的恶意模式。与 BlindBox 相比，这确实是我们论文的关键贡献，因为在我们的协议中，SP 既不知道流量内容也不知道模式签名。该 SP 能够盲目地检测攻击，不知道恶意模式，它是寻找交通。

安全编辑器（SE）角色在我们的协议中也作为*诚实但好奇的*实体进行处理。所有检测签名都包含真实和真实的恶意模式，其主要目的是专门检测和限定网络攻击。这是一个相当合理的假设，因为 SE 不会通过发布错误或误导的签名来损害其声誉。此外，与良性流量匹配的错误签名将导致误报，这可能会降低最终用户所感知的服务质量。但是，SE 很好奇，因为它可能会尝试获取有关流量明文内容的信息。这可能是通过网络上的直接监听，也可能是通过 SP 警报。

勾结 SE 和 SP 使用我们的加密协议将能够安装字典攻击。因此，我们认为 SE 和 SP 在我们的安全模型中不会一起协作。我们认为这个假设是合理的，因为 SE 和 SP 可能都很好奇，但是公开的不诚实行为会在自由市场环境中给他们造成广泛的损害。

串连的发送者和接收者可以使用 SE 和 SP 都无法到达的第三通道来**协商**共享秘密。如果发件人和收件人是恶意的，我们的安全模型不会期望双方合作。我们以受感染的机器人连接到受同一攻击者控制的命令服务器为例。我们可以将这种情况分为两个阶段。首先，攻击者可能会破坏一个良性终端并将其招募到恶意僵尸网络中。其次，被感染的漫游器连接到远程命令并控制服务器是由攻击者为此专门设置的。不幸的是，当使用网络加密时，我们的安全模型仅检测到第一阶段。但是，这也是对所有现有 SSL 检查解决方案的限制，因为受感染的 bot 和命令服务器可能会同意使用秘密编码或加密来隐藏恶意命令，因为它们无法被纯文本检测签名覆盖。

为了简单起见，在本文的其余部分中，我们认为发送者角色是恶意的，而接收者则是诚实的。我们的加密协议在诚实发送者和恶意接收者的上下文中以相同的方式应用。

2.3 一种新的安全模型

基于以上说明，我们现在正式定义安全模型，该模型表示在加密流量上运行的理想入侵检测系统。我们首先对参与者之间的交互进行建模，然后描述检测系统需要满足的三个主要安全属性。

我们考虑一组（未加密的）规则 R 和一个检测算法，表示为 Detect，将（未加密的）流量 *T* 和该集合 R 作为输入。在后文中，我们说该流量是恶意的 iff Detect（T， R）=0。否则，流量是安全的，并且 Detect（T，R）=1。在两种情况下，也可以提供一些辅助信息 aux 作为检测算法的输出。我们还考虑在模型中检测程序可以输出更详细的信息（例如已识别出的恶意模式数量[1]）。

交互模型。入侵流量检测系统由五个主要过程组成，该入侵检测系统由 n 个角色组成，由四个参与者服务编辑器（SE），服务提供者（SP），发件人（S）和接收者（R）播放。

- Setup（输入），输入安全参数 A 后，将生成系统的公共参数 param，以及参与者的潜在键。当一个演员 甲（甲∈{ SE， SP， 小号，-[R}）管理一个加密密钥，我们总是认为有一个密钥对（SK甲，PK甲）其中 SK甲是秘密和只由已知的甲，和 pk甲公开可用。后一个密钥可以为空（在基于密钥的解决方案的情况下）。这样的过程可以由一个演员唯一地执行，或者由几个演员扮演（例如，每个演员 A 可以创建自己的键）。我们认为所有公共密钥现在都包含在参数 param 中。

- RuleGen，上输入参数 PARAM 中，SE 的秘密密钥 SK 中 sE 和一个规则集合 R 来检测恶意流量，输出一组乙的盲法的规则，然后被发送到 SP 。

- 发送 作为输入公开参数 PARAM，潜在的秘密密钥 SK 小号发送者和公钥 PK甲的接收机的[R，和一个流量 T.它输出一个加密的流量 *Ë* 接收机 ř 。

[1] 即使模式本身对于执行检测算法的服务提供商来说仍然是未知的。例如，在我们的构造中，SP 知道哪个陷门匹配，但不知道底层关键字，有关详细信息，请参见第 5 节。

实验 Exp ^ A （A）（参数，sk s E ， sk R） 设置（1 ^）；

B RuleGen （param ， sk S E，R）；

E A （1 A，param ）；

如果 Detect （param ， E，*B*） = 1，则返回 0;

T 接收（param ，sk R，E）；

如果 Detect （T，R）= 0，则返回 0。

图 1：检测实验

实验 Exp ） ') “” （A; ib■ ”，1};

（param ，sk SE，sk R） 设置（1 ^）；

T o，T i A（1 \ param ）；

如果类型（T o，T i）= 0，则返回 0;

E b Send （param ，T b）；

b ' 甲（E b）；

返回（b = b ' ）；

图 2：流量不可分辨性实验

- 根据输入参数，检测服务提供商的公共密钥 pk SP（如果存在），加密的流量 E 和来自 SE 的盲规则集 B，输出位 *b* G {0,1}，表明基础流量 T 是恶意的（b = 0）或安全的（b = 1）。它还可以返回一些辅助信息 aux，例如匹配的盲法。如果出了什么问题，它将输出一条错误消息±。
- 接收 通过取在输入的参数执行 PARAM ，接收器的秘密密钥 SK R 和加密的流量 E.它输出一个普通的交通 T，或错误消息±。

备注 1.*在执行过程* Setup *以生成公共参数* param *和键之后，我们表示 B* = RuleGen（param ， sk SE，R），*其中 R 是规则集，而 E* = Send（param ，sk S，pk R，T），*其中 T 是流量。通过加密流量的入侵检测系统是正确的*

T =接收（param ，sk R，pk S，E），*而* Detect （T，R）= Detect （param ，E，B）*（包括* aux）。

安全属性模型。如上一节所述，主要有三个安全属性应由这种系统验证。我们将第一个称为检测属性，第二个称为流量不可分辨属性，最后一个称为签名不可分辨属性。

*检测。*非正式地说，检测属性指出，建议的入侵检测系统中的服务提供商必须通过加密流量来检测任何恶意流量（即未加密时被视为恶意流量）。这与安全意识功能有关，并提供了一种保证检测正确性的方法。

更正式地，我们在图 1 中给出了针对对手 *A* 的实验。输入参数后，*A* 输出加密的流量 E，使得 Detect（param ，E，B）= 1（表示为安全），而解密版本 T 是恶意的（即 Detect （T，R）= 0）。

然后，如果对于任何概率多项式时间 A，存在可以忽略的函数 v（A），则表示 *可以检测到*加密流量 n 上的入侵检测系统：

Succ n) A (A) = pr [EX P 在 A = ^ < V (A) 。

*交通难以区分。*流量不可分辨性属性非正式地指出，服务提供商无法学习有关流量的任何信息，除了恶意或安全之外。在这里，我们将重点放在对隐私友好的功能上，即无法访问明文内容。

在传统的不可区分性中，对手选择两个消息，并且应该不能区分挑战者加密了两者中的哪一个。在我们的上下文中，我们存在一个问题，即对手可以选择一种恶意流量和一种安全流量，从而仅通过应用“检测”算法就可以轻松区分挑战者使用的是哪种。此外，该检测算法还可以向 A 提供一些有关攻击类型的辅助信息 aux，以便它可以相应地选择两个消息。然后，使用以下定义介绍流量*类型*的概念。

定义 1 （流量类型）。*令* T o *和* T i *为两个流量，令* R *为一组规则。我们说* T o *和* T 我 *是的*相同类型，*表示为*式（T o，T 我）= 1，*当且仅当*

Detect （param ，T o，R）=检测（param ，T i，R），

包括辅助信息 aux 。

更正式地讲，我们在图 2 中进行了流量不可分辨性实验，针对一个既可以访问 Receive oracle（给定选择的加密流量 E，A 获得相关的普通流量 E）又可以访问 RuleGen oracle（假设 a 如果选择了一组规则 R，则对手会获得 B RuleGen （param ，sk SE，R）。对手
首先选择两个流量 T o 和 T i，如果它们具有相同的类型，则选择其中一个，将 T b 加密并提供给 A。最终，A 必须猜测比特 b。

然后，如果对于任何概率多项式时间 A，存在可以忽略的函数 v（A），则表示加密流量 n 上的入侵检测系统是*无法区分的流量*：

广告< A ed （A） = | 2•Pr J xp? O' = 1] -1 | <v （A）。

*规则不可区分。*最后，规则不可区分性属性非正式地指出，服务提供商无法学习有关规则的任何信息。在这里，我们处理符合市场要求的功能因为它可以确保图案签名的私密性。

同样，我们需要处理这样一个事实，即服务提供商可以创建其选择的任何流量，并利用加密的规则对其进行测试并了解一些信息。这是对签名的暴力攻击，发送大量（随机）流量以猜测规则背后的逻辑。这是对安全解决方案的典型测试，它适用于加密和非加密规则：基于加密流量的入侵检测系统无法抵抗这种攻击，并且

实验\ XP ：，'4"一

b ■ “，1};
（param，sk s E，sk R） 设置（1 A）；
R o，R i A f（1 A，param）；
B b RuleGen（param，sk s E，R b）；
b' 甲克（SK 界，乙 b）；
返回（b = b'）；

图 3：规则不可区分性实验

在模型中不应考虑它。要点是，除了提供给检测算法的输出外，SP 无法学习任何信息。然后，我们的想法是使用高-最小熵属性，该属性非正式地指出对手无法“偶然地”获得规则。更正式地，我们使用以下定义（例如参见[10]）。

定义 2（MIN-ENTROPY）。*概率对手 A* =（A f，A_g）*具有*最小熵 *p*

VAË Ñ VR ER：镨'[R' 甲'F（1 ，B）：R' = R] <2 '。

如果 A *具有* p（A）ew（log A）的*最小熵 p，则* A *具有*较高的最小熵。

图 3 中给出了与规则不可区分性相关的实验，其中对手 A =（A f，A_g）具有高-最小熵（考虑到 A_f 和 A_g 不能相互交流，例如[10]]），可以创建其选择的任何流量。简而言之，对手 A $_f$ 选择两组规则 R o 和 R i，其中一组用于 RuleGen 过程。然后将输出 B_b 提供给 A_g，最后输出位 b。

然后， 如果对于任何具有高最小值熵的概率多项式时间 A =（A f，A_g），存在可忽略的函数 v（A），则说加密的流量 n 上的入侵检测系统是*无法区分规则的*：

进阶^ - ^ A）= 1 2•镨' LXP ™, "' = 1] -1 | <v（A）。

3. 技术贡献

我们在本节中回顾有关应用于加密流量的应用程序级安全功能的相关工作。我们强调与我们的工作最接近的 BlindBox 方法[25]，然后我们详细介绍了与 BlindBox 相比我们工作的主要技术贡献。

3.1 相关工作

相关工作包括近期的多项研究，这些研究针对加密网络流量的上下文中的安全性要求。[16]中的作者对加密流量入侵检测的主要贡献进行了调查。[16]中的重点仅是流量分析，它使用流量的高级功能，例如数据包大小，熵和应用程序标识。应用于这些功能的其他统计信息和机器学习技术能够检测特定的攻击类型，例如扫描尝试[29]，拒绝服务[7]和蛮力[8，9]。这些统计技术大多与加密协议无关。它们不支持任何 DPI 功能，并且以相同的方式应用于加密流量和明文流量。

在[30]中，作者介绍了一个称为 QoS2 的框架，该框架使网络中间盒能够传递加密的内容，以支持面向性能的功能，例如内容缓存。QoS2 扩展了 Web 服务器的功能，使其能够提供混合内容，包括同一连接内的加密内容和明文内容。主要概念是，内容提供者可以将私有内容（通过 HTTPS 发送）与其他通过 HTTP 发送的公共内容分开。QoS2 通过生成所有公共内容的校验和并通过安全的 HTTPS 连接发送这些校验和来防止中间人攻击。仅基于每个连接中的公共内容，网络中间盒将进一步支持内容缓存。因此，QoS2 将自身限制为通过常规 HTTP 传递的公共内容，但私有内容无法进行 DPI 之类的安全操作。恶意提供者因此可以在专用连接中模拟恶意内容，而网络中间盒无法通过 DPI 检测到恶意内容。

在[18]中，作者提出了一种深层数据包过滤协议，该协议利用了软件定义网络范式，以便在加密流量上提供过滤功能。在该协议中，服务提供商（SP ）首先将每个新网络连接需要检查的标头字段通知发送方。然后，发送方和 SP 运行交互式协议来加密检测规则集，其中，发送方输入一个自己生成的密钥，提供者输入该组过滤规则。[18]中的协议有两个主要限制。首先，每个 HTTPS 连接都会对过滤规则加密一次，这使得在大型实际部署方案中很难扩展。二，SP 通知发送方有关将由过滤规则检查的标头字段，这要求 SP 拥有明文形式的对这些规则的完全访问权限。这显然不符合安全市场生态系统的现实约束。

Melis 等。在[20，6]中提出了一种新的解决方案，该解决方案可以在云中实现网络功能的隐私保护外包。[20，6]中的贡献与我们的工作类似，它通过保护敏感安全策略的隐私免受好奇的服务提供商的侵害。但是，它仅适用于专用网络环境。它需要企业网络的出口网关来加密明文流量内容并将其转发到基于云的安全功能。我们的解决方案与众不同，因为它能够以端到端的方式对网络连接的发送方和接收方之间的流量进行加密。它保留了安全策略和网络通信内容的私密性。

与我们最接近的工作是 BlindBox [25]及其扩展程序 Embark，它支持将其外包到云[17]。BlindBox 使用多方计算，例如乱码和遗忘传输，并通过三种不同的检测协议支持 DPI。第一协议使 SP 能够在加密流量中的随机位置搜索模式或关键字。第二种协议通过允许还搜索特定偏移量的模式来扩展第一种协议。最后，第三种协议支持可能的原因解密。它允许 SP 使用前两个协议在检测到可疑关键字时对流量进行解密。为此，它将解密密钥嵌入到用于模式匹配的活板门中，从而导致对流量进行完全解密。然后，SP 便可以运行

在明文流量上使用完整的 IDS，完善了前两个协议的结果。

3.2 BlindBox 的局限性

尽管 BlindBox 解决方案提供了有价值的贡献，从而推动了尊重隐私的入侵检测技术的发展，但是它却受到两个主要限制。第一个是关于可扩展性，第二个是关于网络安全解决方案的市场生态系统的合规性。

BlindBox 实施的加密协议要求 SP 与两个端点进行乱码评估，对于每个 HTTPS 连接一次，对于在流量上测试的每个检测规则一次。实际上，BlindBox 解决方案利用常规的 SSL 握手来生成用于加密的对称密钥，这对于每个 HTTPS 连接来说都是唯一的。因此，SP 将需要与发送者联系以便获得其用于确定性地加密安全编辑器提供的每个检测规则的密钥。这些规则随后将由 SP 存储 HTTPS 整个连接期间用于 DPI 的目的。这极大地增加了每个 HTTPS 连接的建立时间，根据[25]达到了 97 秒，因此在现实世界的限制下使其不切实际。SP 设备上可用的内存空间还需要根据（*a*）要保护的唯一端点的数量，（*b*）每个单个端点的唯一 HTTPS 连接以及（*c*）SP 支持的唯一检测规则的数量成倍增长。。显然，这将限制 BlindBox 在小型网络环境中的使用，包括有限数量的端点和检测规则。

其次，为每个新的 HTTPS 连接加密一次检测规则，要求 SP 以明文形式直接访问这些规则。由于安全编辑器（SE ）非常不愿意与服务提供商共享检测规则，因此这显然不足以与安全市场生态系统中的微妙平衡保持平衡。使用 BlindBox 解决方案，解决此限制的一种可能性是 SP 在每个新的连接设置中与 SE 内联，以便 SE 代表 SP 加密规则。然后，整个加密规则集将由 SE 传递给 SP 为了实现 DPI 服务。尽管从技术角度来看，此解决方案似乎是可行的，但由于需要在 SE 和 SP 之间通过网络共享整个加密规则集，因此它将在连接建立期间增加巨大的延迟。

最后，值得注意的是，[25]要求的完整 IDS 功能可与嵌入式解密密钥配合使用，该密钥在匹配的情况下会显示出来。但是，此技术不允许 SP 具有完整的 IDS 功能。实际上，不会对整个流量评估正则表达式，而只会对已经可疑的通讯进行评估。这样，它仅有助于限制误报的数量，而不能减少误报的数量。

Embark 系统[17]不处理这些限制。它仅旨在扩展最初的 BlindBox 解决方案，并具有将网络中间盒安全外包给云的能力。

3.3 建议的解决方案

我们的系统为所有用于以下目的的 HTTPS 连接*仅*加密*一次*用于检测的恶意模式通过 SP 交付。与 BlindBox 相比，这是本文的基本贡献，因为以下两个原因。首先，由于 SP 将不再需要延迟连接建立，直到它对用于检测的整个恶意模式进行加密为止，它大大减少了几个数量级的连接建立时间。它还减少了 SP 执行 DPI 所需的内存空间，这仅取决于检测规则的数量，而不取决于端点的数量和/或并发 HTTPS 连接的数量。其次，它可以将规则加密外包给 SE，它将能够加密（又称保护）其检测规则，然后仅将其加密后传递给 SP 。

可以肯定的是，使用相同的加密规则集进行检测可能会揭示不同连接上匹配模式的相等性。通过比较大型网络连接上的匹配规则（类似于暴力攻击），安全提供程序可能能够恢复某些检测模式，这可能会部分损害市场遵从性。尽管这种暴力攻击至少在理论上是可能的，但它们同时适用于加密和明文网络连接。因此，我们不会将它们视为对我们解决方案的限制，因为它们并不是我们的加密协议固有的。而且，

为此，我们的解决方案利用了基于公钥密码学的可解密可搜索加密（DSE）方案[12]。我们的协议使 SE 能够生成一个公钥对（pk_R，sk_R）和一个活板门密钥。SE 使用陷门密钥来加密恶意检测模式并获取陷门。这些进一步发送到 SP，以应用 DPI。使用相同的 DSE 方案[12]，发送方使用接收方的公钥对流量进行加密，SP 使用 SE 生成的陷阱门对加密的流量执行 DPI。。尽管使用公钥加密会增加接收器的解密开销，但在第 5 节中进行的广泛评估表明，在 HTTPS 连接短的情况下，这是一个相当合理的价格，这对于大多数流行的网站都是常见的在网上。

4. 我们的 DSE 解决方案

我们的解决方案基于所谓的可解密可搜索加密（DSE）加密工具[12]这是一种基于配对的方案。在本节中，我们简要介绍双线性映射和 DSE 然后描述使用 DSE 协议对加密流量进行入侵检测的方式。

4.1 双线性环境

首先，让我们回顾一下双线性环境的概念。

令 G_1，G_2和 G_t为大素数 q *的*三个有限乘法阿贝尔群。令 g_2是 G_1 的生成器，g_2是 G_2的生成器。我们假设存在一个不对称的双线性图 e：$G_1 \times G_2 \to G_t$，因此对于所有 a，$b \in Z_q$：

1. $e(g_1, g_2) = e(g_1, g_2)^{ab}$，
2. $e(g_1, g_2) = 1$iff $a = 0$ 或 $b = 0$，

3. e（-，•）是可有效计算的。

在续集中，元组（q，G_1，g_1，G_2，g_2，G_t，e（-，•））被称为双线性环境。

4.2 可解密的可搜索加密

正式定义。可解密的可搜索加密方案是带有关键字搜索（又名 PEKS）的公钥加密，另外还有特定秘密密钥 sk 的所有者解密密文的可能性。非正式地，关键字搜索属性指出给定密文 *c* 和关键字 w，测试密文是否与关键字匹配是可行的。此功能对于解密密钥的所有者而言是显而易见的，但也可以以关键字特定的陷门 *T（w）*的形式转移给任何人。DSE 方案由以下过程组成。

- KeyGen 是密钥生成算法，它以安全参数 A 为输入，并输出公共密钥 pk，解密专用密钥 sk 和活板秘密密钥 tk。
- Enc 是加密过程，输入关键字 w 和公共密钥 pk 作为输入。它输出与 w 和 pk 有关的密文 c。
- TrapGen 是陷阱门的生成，该陷阱门将关键字 w 和陷阱门密钥 tk 作为输入，并输出陷阱门 T（w）。
- 测试是一种测试算法。它以密文 c 和活板门 T（w）作为输入，如果 c 是 w 的加密，则输出 1，否则输出 0。
- Dec 是解密过程。它以密文 c 和私钥 sk 作为输入。它输出相关的关键字 w。

如[12]中所述，DSE 方案应验证针对所选密文攻击（CCA）的特殊形式的不可区分性。此属性与一个实验相关，这样，给定公钥 pk，对手就会输出两个不同的关键字 w_0 和 w_1 并且在随机选择了 *b* G {0,1}位之后，挑战者将 c_b 作为密文输出与 w_b 有关。最终，如果 *b′* = b，则对手输出 *b′*并获胜。在整个实验过程中，对手可能将查询 w G {w_0，w_1}发送给活板推导 oracle，并且查询 c = c_b 解密解密。我们说，如果对于每个合法对手来说，DSE 满足*选择明文攻击下的不可区分性*，那么赢得该实验的优势微不足道。

DSE 方案。在[12]中，作者提出了以下有效的构造。令*（q，G_1，g_1，G_2，g_2，G_t*，e（。，。））为双线性环境，令 *F，G，H* 为三个哈希函数，每个哈希函数均建模为随机预言。

- KeyGen （1$^{\lambda}$）。此过程随机地选择 *X*，*X* · G ∧ $\check{Z}_q$ 并计算 $\ddot{y}$ =克^{F} 和 y' = G ^{F}。我们有 sk = *x*， tk = x ·和 pk =（y，y'）。
- Enc （w，pk）。此过程选 $\check{r}$ $\dot{g}$ $\check{Z}_q$ 和计算 c我= G ^{F}，（S我，S 2）= *G（*γ ^{R}），C 2 = S我®W，C 3 =克^{F}，$\hat{u}$ = E（Y $^{\prime}$，F（w））和 c_4 = H（u）+ r（mod q）。密文为 c =（c_{J}，c_2，c_3，c_4）。
- TrapGen （w，tk）。此过程输出活板门 T（w）= F（w）$^{\prime}$。
- 测试（c，T（w））。该过程首先计算值 u = e（c_3，T（w））和 r = c_4 -H（u）（mod q）。如果 c_{j} = G $^{[}$，则返回 0，否则，它计算 S =Y ^{R}，（S_{J}，S_2）= G（S）和 W = C $_2$ ®小号 $\hat{\text{j}}$。。如果 c_3 = g ^{f2}，则返回 0。否则返回 1。
- Dec （c，sk）。这个过程计算 S = C ^{F}，（一个或多个学家，S_2）= G（S）和 W = C $_2$ ®小号 $\hat{\text{j}}$。。如果 c_3 = g ^{f2}，则返回±。否则，它计算 $\hat{u}$= E（Y $^{\prime}$小号，F（w）的）中，r = C_4 - H（U）（MOD Q）并检查是否 Ç $\hat{\text{j}}$ = g ^{f}。如果满足此条件，则返回 w。否则返回±。

4.3 概述

基于上述 DSE 工具，我们现在提供一种用于通过加密流量检测恶意内容的解决方案。例如，在 IDS 负责过滤掉恶意内容的情况下，应以服务提供商（SP）可以检测到恶意内容的方式来建立协议，并且如果流量没有受到损害，则 SP 应该能够在没有获得任何信息的情况下将加密的业务转发到接收器 R。

首先，我们认为发送方 S 和接收方 R 之间的流量是使用 DSE 加密的（尤其是上一节中所述的修改版本，如下所示）。这为我们提供了主要的所需属性：流量的机密性，以及使用相等性测试检测恶意流量的可能性。

足够的证券市场。这样，我们还获得了一个附加属性：服务提供程序（SP 不知道搜索的模式，因为它仅知道活板门 T（w），而不知道关键字 w 本身。但考虑在给定的有效实例化时[12]，我们不得不面对的是，当这个问题的试验程序是成功的，它也允许输出的匹配关键字，如一个可以计算瓦特= C 2 ®秒。然后，我们修改以上方案以实现我们的目标。在下面的协议中，我们将 c_4 计算为 c_4 = H（u）+ a（mod q），其中 $G_{z\cdot}$ 属于 SE 给定的公钥。使用这种技巧，即使在找到匹配项的情况下，SP 也无法获取有关关键字的信息，正如我们在安全性证明中所看到的。因此，与 BlindBox 解决方案相反，我们符合当前的安全市场。

可扩展性。为了实现我们的第二个目标，我们使用了另一个技巧，即观察活板门和解密密钥在 Furh-Paillier 有效构造中是完全独立的[12]。然后，可以独立计算与“加密”阶段有关的公钥 $y_{\text{和}}$ 与“测试”阶段有关的公钥 y'。结果是，安全编辑器 SE 可以计算自己的活板门密钥 tk = x $^{\prime}$（以及相应的公共密钥 y' = g ^{f}），而无需知道解密密钥 sk = x（以及相应的公共密钥 y = g ^{f}）。）。然后，我们可以管理几个密钥对（x_{j}，y_{j}）$_{\text{j}}$，使得 x_i G_R Z_q 和 y_{i} = g ^{f2}：每个接收器 R_{i}。通过这种方式，我们可以轻松获得可伸缩性，因为 SE 仅发布了一组陷阱门，以允许 SP 检测到任意数量的端点所需的恶意流量。同样，这在两个方面都比 BlindBox 解决方案好得多

设置的时间复杂度以及 SP 的空间复杂度。

协议概述。我们的解决方案 BlindIDS-DSE 利用 DSE 的这一双重特性，使用[12]中提到的加密块，对加密数据建立了深度数据包检查协议。在接收或生成流量之后，并且在加密之前，发送者（S）将令牌化算法应用于流量。有两种令牌化算法在检测时会产生不同的性能。标记化算法产生固定长度（基于窗口的标记化）或可变长度（基于定界符）的关键字，如[25]所示。

BlindIDS-DSE 协议由四个代理组成，每个代理实现在检测系统中起作用的不同模块。该协议如下运行。

- **系统设置**程序（Setup）

  - RG 运行 DSE 密钥生成算法来生成秘密暗门关键 SK 小号Ë，以及相关的公共密钥 PK 发。

  —独立地，每个接收器 R 运行 DSE 密钥生成协议，以生成公共密钥对（pk R，sk R）。

- **规则生成**（RuleGen）

  —SE 使用带有固定长度关键字的陷门生成过程，以便为与攻击相关的每个关键字 w 生成陷门 $T(w)$。

  —SE 将活板门发送给 SP 进行检测。

- **发送准备**（发送）

  —首先，SP 与 S 和 R 建立连接。

  —S 从流量中产生固定长度的令牌，并使用特定的接收器 R 的公钥加密每个令牌，并在令牌中附加其在有效负载中的位置（以使反向令牌化算法能够重构流量），使用 DSE 加密算法 Enc。

  -小号将加密的令牌-[R。

- **检测**（检测）
  SP 对包含陷阱门的树数据结构中的每个加密令牌和条目运行相等性测试。如果规则中的所有关键字都匹配，则 SP 的检测模块将输出 0，否则将输出 1。因此，SP 将丢弃数据包，或者将令牌发送给 R，具体取决于 SP 实施的安全策略。

- **验证和数据包重构**（接收）
  R 使用解密过程 Dec 接收加密令牌的集合并运行反向令牌化算法以重建消息。基于发送方实施的令牌化过程，可以将每个解密关键字的位置与加密令牌一起转发，以使接收方能够反转令牌化算法并以明文形式重构流量。

4.4 详细说明

令$(q, G_1, g_1, G_2, g_2, G_t, e(\cdot,\cdot))$为双线性环境，令 F，G，H为三个哈希函数（建模为随机预言机）。

- **设置**。系统设置包括两个独立的密钥生成步骤。

  -该 SE 执行 DSE。注册机（1 个参）程序，仅保持陷阱生成密钥 TK = X”
  它出版 PK 发=克 j，具有一定的随机字符串沿一个 g ž j。

  —接收器 R 还执行 DSE。KeyGen（1λ）获得其秘密解密密钥 sk R = x Z_q。
  它发布了相关的公钥 pk R = g_j。

  -用于接收器的公共密钥-[R 是一对 PK 接收= (P 钥 SE，p 钥-[R)。

- RuleGen。对于 在业务中要搜索的每个关键字 w_i，SE 执行 DSE 方案的 TrapGen（w i，tk）。返回 T_i = F（wi）。然后，SE 将陷阱集 $T = \{T_1, .. T_l\}$发送到 SP。

- **发送**。

  —S 将流量分成令牌 $t_1, \ldots, t_n$。

  —对于每个令牌 t_i，S 在 Z q 中均匀地随机抽取 r_i并执行我们修改后的 DSE 方案的 Enc（t i，pk R）。此过程计算以下内容：

$$c_{i,1} = g^{r};$$
$$(s_{i,1}, s_{i,2}) = G(pk_R^{r});$$
$$c_{2,i} = 小号_1, 我 \ddot{O}^{我}$$
$$c_{3,i} = 小号_{2} \oplus 我$$
$$U_i = e(pk_1^{r}, F(t_i));$$
$$Ç_{我} = H(U_{我}) + 一个模 Q 值。$$

  对于每个令牌吨密文我是四联 ç我-(c 1，我，c 2，我，c 3，我，c 4，我)。

  —加密的流量 E 是每个令牌的密文的集合。小号 发送 E 至 ř。

- **检测**。当服务提供者拦截 S 和 R 之间的加密流量 E 时，它将执行过程 DSE。在每个密文上测试（c i，T_j）以查找匹配的签名。所述测试过程包括以下步骤。

  -该 SP 首先计算 u 对应的值我= $ë(C_{3,我}, T_j)$，然后计算一个 ' = $c^{\wedge}{}_{,我}$- $H(ú_{我})$模 Q 值。

  —如果 a = a '，则返回 1。这意味着 t i 和 r j 不同，因此签名与流量不匹配。否则，它返回 0。

  如果该过程返回 0，则服务提供商将生成警报。它还可以返回辅助信息 aux = T i，它是匹配的活板门。否则，流量被转发到[R。

- **接收**。对于每个密文，R 执行修改后的 DSE 方案的 Dec（c i，sk R），该过程由以下步骤组成。

— R 计算 $s_{,}$ = cj ^并使用 G 检索对（s i, ，S 2, ）。然后，它计算*吨*= C $_{2,}$ $_{我}$®小号 $_{1,}$ $_{我}$。

—如果 c $_{3,}$ $_{i}$ = g 1 ^{2,1}，则过程返回±。

—否则，它计算出 u $_{i}$ = e（pk ||%，F（t $_{i}$））。

—如果 H（u_{i}）— c_{4} y = a，则返回±。

—返回 t $_{i}$。

尽管我们的协议主要用于模式匹配，但是我们可以使用与 BlindBox 相同的密钥嵌入技术来评估可疑流量的正则表达式。有关详细信息，请参见[25]。

4.5 安全性

按照安全模型，我们有以下三个定理，其中 n 是加密流量上的入侵检测系统。证明在附录 A 中给出。

定理 1.*只要陷门生成函数中没有冲突，我们的方案 n 是可检测的。*

我们在两个假设下证明了不可区分性：计算 Diffie-Hellman 问题（CDH 和 GDDHE 假设，后者在[11]中引入。为了简单起见，我们给出 GDDHE 假设的一个非正式版本。

定义 3 （CDH）。*令 g，g^{a}，g^{b} 为三个元素*

概率多项式时间的对手 G 在计算 g_{ab} 时具有可忽略的概率。

定义 4 （（（（P，Q，f）-GDDHE））。*设 s，n 为 pos 正整数和 P，Q* G F $_{p}$[X $_{1}$，...，X $_{n}$] $_{s}$ *是* F $_{p}$ *上 n 个变量多项式的两个 s 元组。令 f* G F $_{p}$[X $_{1}$，...，X $_{n}$] *一个与 P 和 Q 线性独立的多项式。给定 H（x_{i}，...，x_{n}）=（g_{P}（x_{1}，g_{Q}（x_{1}，...，x_{n}））* $_{e}$ G 1 X Ç *小号 2 和 T* g ^ g ^ $_{T}$，*概率多项式时间对手具有可忽略的概率成功地决定如果 T* = G $^{F(x_{1}, \ldots, x_{n})}$。

定理 2。*在随机预言模型中，在 CDH 和 GDDHE 假设下，我们的方案 n 在流量上是无法区分的。*

定理 3。*在随机预言模型中，对于高最小熵的规则，我们的方案 n 是不可区分的。*

5. 实验

本节介绍了我们的实验设置，包括实施细节和对加密协议的评估。它旨在评估我们解决方案的功能和性能，并将其与类似水平的解决方案（例如 BlindBox）进行比较。

5.1 实施

我们在具有 64 位 Linux OS 的 E5-1620 CPU 和 4 个内核，运行于 3.70GHz 的 Intel®Xeon®上实现了协议。我们在它们的默认 254 位 Barreto-Naehrig 曲线上使用了用 C 编写的最佳 Ate 库[21]。我们使用[21]提供的 Java API 和 Java Standard Edition u112 构建了 Java 8 中的协议。我们使用 SHA 256 作为哈希函数，并使用 Java SecureRandom 类生成随机数。

数据集	参赛作品	支持的条目	%
恶意软件阻止列表[1]	1,250	1,250	100%
网址黑名单[3]	4,546,341	4,546,341	100%
亚拉规则[4]	256	198	77.3%
鼻息通讯。规则[2]	3,467	2,606	75%

图 4：使用我们的解决方案评估的数据集

5.2 评估

我们在本段中评估解决方案的功能和性能。功能评估利用了我们协议使用标准检测签名（例如恶意软件阻止列表，内容过滤和数据泄漏保护，父母控制以及通用 Snort IDS 规则）检测攻击的能力。性能评估利用了客户端-服务器端以及服务提供商提供的检测设备的开销。它还利用了我们解决方案在实际网络配置和流量速率下扩展的能力。

5.2.1 功能评估

总体而言，我们的解决方案提供了与 Blindbox 相似的功能属性，同时保留了与服务提供商有关的检测规则的私密性。为了验证我们解决方案的检测功能，我们引用了多个公共数据集。如图 4 所示，这些数据集提供了广泛的功能，例如恶意软件域列表[1]，通过检测非法广告，色情和暴力或恐怖内容[3]进行家长控制，使用 Yara 规则进行恶意软件检测

[4] 和 Snort 社区规则[2]。

首先，我们评估了解决方案实施每个单独数据集提供的检测规则的能力。恶意软件阻止列表数据集[1]提供提供恶意软件感染内容的完全限定 URL 的公共列表。URL 黑名单[3]数据集具有类似的结构，但涵盖范围更广的非法内容，包括父母控制。这两个 URL 数据集都可以通过比较来应用，并且与传出 HTTP 请求的 URL 标头字段完全匹配。我们的解决方案同时使用基于定界符和基于窗口的标记，可以将这些 URL 与加密流量完全匹配，从而提供 100％的检测率。Yara 规则数据集[4]具有不同的结构。它提供了 DPI 规则，用于搜索十六进制字符串，文本关键字，和通用正则表达式来表征流量负载中被恶意软件感染的内容。公开的 Yara 文档提供了创建 Yara 规则的一般准则以及增强检测性能的最佳实践。在此范围内，除非必要，否则它不赞成使用完整的正则表达式。正则表达式天生就很慢，因此 Yara 项目建议尽可能使用字符串搜索以及跳转和通配符替换它们。我们的 Yara 数据集包含 256 个不同的规则，其中 198 个规则（占 77.3％）仅包含可以使用我们的解决方案通过加密流量完全检测到的关键字搜索。其余规则包括需要直接访问流量的正则表达式。这些规则确实是我们使用解密属性的解决方案所支持的，当 SP 包含使用关键字搜索检测到的恶意内容时，它允许 SP 解密可疑连接。Snort 规则[2]具有与 Yara 规则相似的属性，因此我们使用 Snort 规则获得的结果几乎与 Yara 规则相同。使用我们的 DSE 原型

col，我们能够评估 3,467 个初始 Snort 规则（占 75％）中不包含完整正则表达式的 2,606 个规则。我们的解决方案使用嵌入的解密属性为其余规则提供支持，与 Yara 规则相同。

其次，我们评估了我们检测公共数据集中检测规则所涵盖的所有攻击和恶意内容的能力。由于我们的 DSE 协议将规则关键字与加密的流量令牌完美匹配，如第 4.5 节所述，我们检测这些攻击的能力仅取决于令牌化策略，与 BlindBox 解决方案相同。为了证实这一假设，我们使用相同的 ICTF 数据集[28]进行了与 BlindBox 论文[25]中相同的实验。此数据集包括*捕获标记*期间收集的网络跟踪练习，包括多个团队，每个团队的任务是维护一套服务，以使它们在整个比赛中保持可用并且毫不妥协。我们的实验结果证实了我们最初的假设。我们实现了几乎与 BlindBox [25]相似的检测精度（因为此实验，我们的 DSE 协议使用应用于相同数据集的相同标记化策略），包括 96.5％的攻击关键字和 98.3 的攻击规则已经用 Snort 检测到了。

5.2.2 *绩效评估*

本节从不同参与者的角度评估我们解决方案的性能。首先，当使用我们的 DSE 协议对数据进行标记和加密时，它利用了发送方和接收方的开销。其次，它利用了 SP 的开销，包括实现我们的检测协议所需的时间和内存。

我们的性能结果（包括标准 SSL 检查技术的基准测试和 BlindBox 解决方案[25]）在图 5 中进行了总结。总体而言，与 BlindBox 相比，我们的解决方案与 BlindBox 相比，其存储空间大幅度减少了 6 个数量级。DPI 设备上的必需配置，使其更接近实际配置。它还将新 HTTPS 连接的建立时间减少了 3 个数量级。我们的解决方案中的建立时间不像 BlindBox 中那样取决于检测规则的数量，因此它类似于标准 HTTPS 连接的连接时间。另一方面，由于使用了公钥加密协议，因此与 BlindBox 相比，我们的解决方案实现了较低的数据加密性能。使用我们的解决方案 发送方将需要 27 毫秒的时间来加密 1500 字节 MTU 的网络数据包的内容。使用我们的解决方案加载一个典型的网页（例如 CNN）平均需要 2.3 秒钟，而使用 BlindBox 加载该网页则需要将近 97 秒钟（主要是因为连接建立时间）。为了减少新 HTTPS 连接的总体设置时间，我们认为数据加密的额外开销是合理的代价。因此，我们的解决方案更适合短时和中等寿命的连接，Internet 上大多数标准网页就是这种情况。在本节的其余部分，我们将详细讨论所有这些发现。而使用 BlindBox 加载同一页面大约需要 97 秒的时间（主要是由于连接建立时间）。为了减少新 HTTPS 连接的总体设置时间，我们认为数据加密的额外开销是合理的代价。因此，我们的解决方案更适合短时和中等寿命的连接，Internet 上大多数标准网页就是这种情况。在本节的其余部分，我们将详细讨论所有这些发现。而使用 BlindBox 加载同一页面大约需要 97 秒的时间（主要是由于连接建立时间）。为了减少新 HTTPS 连接的总体设置时间，我们认为数据加密的额外开销是合理的代价。因此，我们的解决方案更适合短时和中等寿命的连接，Internet 上大多数标准网页就是这种情况。在本节的其余部分，我们将详细讨论所有这些发现。我们的解决方案更适合短时间和中等寿命的连接，Internet 上大多数标准网页就是这种情况。在本节的其余部分，我们将详细讨论所有这些发现。我们的解决方案更适合短时间和中等寿命的连接，Internet 上大多数标准网页就是这种情况。在本节的其余部分，我们将详细讨论所有这些发现。

发送者和接收者的开销。发送方和接收方的开销涵盖了在网络上共享数据之前进行连接建立和加密的时间。

*连接建立时间。*由于我们使用的公共密钥加密协议最初不需要 SSL 握手，因此建立时间与 SSL / TLS 连接相同，如图 5 所示。与 BlindBox 相比，使用我们的解决方案的建立时间并不取决于根据 SE 提供的检测规则。我们的 DSE 协议使 SE 能够生成与其公钥对（pk_R，sk_R）相关联的陷门密钥。SE 仅使用一次 trapdoor 密钥对每个新检测规则中的关键字进行加密。活板门由 SE 进一步传送给 SP，没有发送者和接收者的参与。因此，与使用 3000 种检测规则的 BlindBox 的设置时间相比，我们的解决方案减少了六个数量级。更有趣的是，与标准 SSL / TLS 协议相比，它不增加连接设置的开销。

*数据加密时间。*但是，这次使用我们的解决方案比使用标准 SSL / TLS 和 BlindBox 的时间更长。对于流量中包含的每个令牌，它对应于使用我们的 DSE 协议对该令牌进行加密的时间，对于 128 位长的令牌，我们将其平均评估为 729＾s。数据加密的开销主要归因于 DSE 协议的使用，该协议使用公钥加密，而不是 SSL / TLS 和 BlindBox 中使用的对称密钥加密。为了更好地评估对最终用户体验的影响，我们加载了多个受欢迎的网站，例如 CNN，Facebook，Twitter，BBC 和美国银行，并根据页面加载时间评估了平均开销， SSL / TLS 和 BlindBox。我们的实验结果总结在图 6 中。与标准 HTTPS 协议相比，我们的解决方案因其较长的数据加密时间而增加了可观的开销。使用 BlindBox 时，连接建立时间非常长，因此可以快速补偿我们解决方案的数据加密时间。总体而言，使用我们的解决方案的平均加载时间对于中小型网页仍然可以接受，但是对于提供大量内容的网站而言，平均加载时间会大大增加。

服务提供商的开销。我们根据检测规则的数量和网络连接的大小，通过测量所需的可用内存空间以及执行检测所需的时间来评估 SP 的开销。

*检测时间。*与 BlindBox 相比，使用我们的解决方案的唯一重要开销是 DPI 设备上的检测时间。使用 3000 个规则的规则集，每个规则平均包含 3 个活板门令牌（总共近 1 万个令牌，与[25]中用于评估 BlindBox 解决方案的实验设置相同），大约需要 74 秒在 SP 在加密的数据包上应用 DPI。与使用 BlindBox 进行连接设置的 97s 相比，我们的解决方案仍将总体开销（包括设置和检测时间）减少了 25％。但是，对于大型网络中的实时入侵检测，它仍然不能很好地扩展。这主要是因为我们的 DSE 协议的测试程序。它使用加密操作针对网络流量中的每个密文，对检测规则集中的每个陷阱门进行测试。因此，它不同于 BlindBox 实现的“检测”过程，后者在加密规则和加密令牌之间进行完美匹配。

角色	描述	SSL 检查	盲盒	我们的解决方案
发件人/收件人	设置（1 个关键字） 设置（3K 规则） 加密（128 位） 加密（1500 字节）	73 毫秒 73 毫秒 13ns 3 ^ s	588ms 97s 69ns 90 ^ s	73 毫秒 73 毫秒 729 ^ s 27 毫秒
服务提供商*（检测时间）*	1 条规则，1 个令牌 1 条规则，1 个数据包 3K 规则，1 个令牌 3K 规则，1 个数据包	不适用不适用不适用 不适用	20ns 5 ^ s 137ns 33 ^ s	691 ^ s 41.3 毫秒 700 毫秒 74 秒
服务提供商*（RAM 使用情况）*	1 条规则，1 条连接 3K 规则，1 条连接 1 个规则，100 个连接 3K 规则，100 个连接	不适用不适用不适用 不适用	1.75 兆字节 5.12GB 175MB 512GB	0.2KB 0.58 兆字节 0.2KB 0.58 兆字节

图 5：我们的解决方案在连接建立和检测期间的性能，以及使用标准 SSL 检查技术和 BlindBox 解决方案进行基准测试

网站	尺寸	HTTPS	盲盒	我们的解决方案
有线电视新闻网	131KB	0.073	97.008	2.373
Facebook	74KB	0.073	97.004	1.073
推特	284KB	0.073	97.017	5.073
英国广播公司	196KB	0.073	97.011	3.573
蟒蛇	74KB	0.073	97.004	1.073

图 6：一些流行网站的加载时间（以秒为单位）

但是，通过消除发送方建立连接的开销，并将此开销部分转移到 SP 端的检测过程，与 BlindBox 相比，我们的解决方案具有明显的优势。首先，SP 可以使用其他试探法，例如域或 IP 信誉，以识别可能的可疑流，然后仅检查到危险目的地的加密连接。而且，SP 可能使用负载平衡在多个服务器上分配计算，这对于发送方可能不可行。我们的解决方案也非常适合在调查和入侵后取证过程中离线使用。尽管使用 BlindBox 至少在理论上可以支持此功能，但是 SP 将需要存储和管理为每个 HTTPS 连接生成的乱码。与本文的贡献相比，这增加了可观的开销。我们的解决方案使 SE 对于每个新的检测规则仅生成一次陷阱门。SP 进一步应用了这些活板门 DPI 设备检查的所有加密连接。最后，我们的解决方案比 BlindBox 更适合用于调查和入侵后取证，还因为它允许 SP 追溯测试 SE 新提供的检测规则。这些规则可能会捕获零时攻击，但在发生加密连接时仍未知。该 SE 可以使用它的陷阱门钥匙确实加密这些新规则，并将其交付给 SP。BlindBox 不支持此功能，因为它要求发送者准备一个乱码并将其发送到 SP，这在入侵后取证的情况下确实不可行。

内存使用情况。尽管使用我们的 DSE 协议进行 SP 的检测时间要比使用 BlindBox 进行检测的时间长得多，但是我们的方法大大减少了 DPI 设备上需要可用的内存空间。这主要是因为用从检测规则中的恶意关键字派生的通用活板门代替了 BlindBox 中使用的乱码。实际上，BlindBox 中使用的乱码电路是由发件人为每个罪过准备的gle HTTPS 连接。每个乱码电路的大小为 599KB，在整个连接过程中都需要由 SP 存储。因此，DPI 设备上所需的内存空间将相对于规则数量和并发 HTTPS 连接数量线性增长。如图 5 的表所示，将存储 100 个并发 HTTPS 连接和 3,000 条检测规则的乱码电路所需的存储空间评估为 512GB RAM。

但是，使用我们的解决方案，SE 对于每个检测关键字仅生成一次陷门。SP 将陷门进一步应用于 DPI 设备检查的所有加密连接。每个活板门的唯一大小为 508 位，这又不取决于加密密钥的大小。所需的存储空间不再取决于并发连接的数量，而仅取决于检测规则的数量，这与用于纯文本入侵检测的所有 DPI 设备相同。在 100 个并发 HTTPS 连接和 3,000 条检测规则的情况下，我们的 BlindIDS 解决方案仅需要 0.58MB RAM。

6. 结论

在本文中，我们介绍了 BlindIDS，这是一种直接在加密流量上运行深度数据包检测（DPI）的新系统。我们基于安全模型正式介绍了我们的解决方案，该安全模型表示基于加密流量的理想入侵检测系统。然后我们提供了适当的安全证明，以验证系统的主要功能。据我们所知，BlindIDS 是第一个弥合网络安全性和隐私之间的差距，同时又保持安全市场生态系统中微妙平衡的系统。它使安全编辑者和服务提供商可以安全地协作，以提供增值的安全服务，该服务还可以保护最终用户数据的机密性。我们的解决方案对所有人都有益：用户将保留自己的隐私安全编辑人员将能够保护其独特的攻击特征，服务提供商将能够提供入侵检测服务，而不会影响最终用户流量的隐私。我们制作了 BlindIDS 的原型实现，并进行了广泛的评估，以评估该解决方案的功能和性能。我们的实验表明，与类似的现有解决方案（例如[25]）相比，BlindIDS 将连接建立时间和在安全设备上执行 DPI 所需的资源都提高了几个数量级。我们制作了 BlindIDS 的原型实现，并进行了广泛的评估，以评估该解决方案的功能和性能。我们的实验表明，与类似的现有解决方案（例如[25]）相比，BlindIDS 将连接建立时间和在安全设备上执行 DPI 所需的资源都提高了几个数量级。我们制作了 BlindIDS 的原型实现，并进行了广泛的评估，以评估该解决方案的功能和性能。我们的实验表明，与类似的现有解决方案（例如[25]）相比，BlindIDS 将连接建立时间和在安全设备上执行 DPI 所需的资源都提高了几个数量级。

7. 参考

[1] 恶意软件域列表。https://www.malwaredomainlist.com/mdl.php，2016 年。

[2] 喷鼻息。https://www.snort.org/downloads/，2016 年。

[3] 网址黑名单。http://www.urlblacklist.com/?sec=home，2016 年。

[4] Yara 规则存储库。https://github.com/Yara-Rules/rules，2016 年。

[5] M. and Markets。威胁情报安全市场的解决方案-到 2020 年的全球预测。在 *MarketsandMarkets 报告 TC 3591 中，* 2015 年。

[6] HJ Asghar，L。Melis，C。Soldani，ED Cristofaro，MA Kaafar 和 L. Mathy。Splitbox：实现高效的专用网络功能虚拟化。在*关于中间盒和网络功能虚拟化的热门主题的研讨会上，*第 7-13 页，2016 年 8 月。

[7] M. Augustin 和 A. Balaz。尽早识别加密应用程序的入侵检测。在 *IEEE 智能工程系统（INES）会议上，* 2011 年 6 月。

[8] M. Barati，A。Abdullah，R。Mahmod，N。Mustapha 和 NI Udzir。使用遗传算法为加密流量中的 ID 进行特征选择。在*国际计算和信息学会议（ICCI）上*，第 279-285 页，2013 年。

[9] M. Barati，A。Abdullah，NI Udzir，M。Behzadi，R。Mahmod 和 N. Mustapha。云环境中安全外壳流量中的入侵检测系统。在*《计算机科学杂志》上*，2014 年第 10 卷。

[10] 贝拉雷（M. Bellare），菲施林（M. Fischlin），奥尼尔（A. O'Neill）和里斯滕帕（T. Ristenpart）。确定性加密：定义等价和无随机预言的构造。在*密码学进展- CRYPTO 2008 年*的成交量 5157 *在计算机科学讲义*，360-378，2008 页。

[11] D. Boneh，X。Boyen 和 E. Goh。具有恒定大小密文的基于分层身份的加密。在 *密码学的进展中-2005 年 EUROCRYPT，第二届年度密码技术理论与应用国际会议，丹麦奥胡斯，2005 年 5 月 22 日至 26 日，会议记录，*第 440-456 页，2005 年。

[12] T. Fuhr 和 P. Paillier。可解密的可搜索加密。在*《可*证明的*安全》*（Provable *Security*）*中*，第 4784 卷，第 228-236 页，2007 年。

[13] R. Holland，S。Balaouras 和 J. Blackborow。网络威胁情报市场的状况。在 *Forrester 报告中*，2015 年。

[14] L.-S. Huang，A。Rice，E。Ellingsen 和 C. Jackson。在野外分析伪造的 ssl 证书。在 *IEEE 安全与隐私研讨会上*，2014 年。

[15] J. Jarmoc。SSL 拦截代理和传递信任。在《*黑帽欧洲》中*，2012 年。

[16] T. Kovanen，G。David 和 T. Hamalainen。调查：加密流量中的入侵检测系统。在*物联网，智能空间和下一代网络和系统中*，*LNCS 的*9870 卷，第 281-293 页，2016 年。

[17] C. Lan，J。Sherry，RA Popa，S。Ratnasamy 和刘 Z. Embark：将中间盒安全地外包到云中。在 *Usenix NSDI 中*，2016。

[18] Y.-H. 林淑华 沉明辉 杨丹妮 杨和 W.-T. 陈。在软件定义的网络中对加密流量进行隐私保护的深度包过滤。在 2016 年 *IEEE 通信大会（ICC）中*。

[19] R. McMillan 和 K. Pratap。安全威胁情报服务的市场指南。在 *Gartner 报告（G00259127）中，* 2014 年。

[20] L. Melis，HJ Asghar，ED Cristofaro 和 MA Kaafar。私有处理外包网络功能：可行性和构建。在 *ACM 国际软件定义网络和网络功能虚拟化安全研讨会上，*第 39-44 页，2016 年 3 月。

[21] M.UNSUNARI。在 Intel Haswell 处理器上快速实现 bn 曲线上的最佳配对。密码学 ePrint 存档，2013 / 362，2013 年报告。http：//eprint.iacr.org/2013/362。

[22] B. Mukherjee，LT Heberlein 和 KN Levitt。网络入侵检测。在 *IEEE Network*，第 8 卷，第 26-41 页，1994 年。

[23] P. Paganini。法国政府的 Anssi 负责针对 Google ssl-tls 的交易。在《*安全事务》杂志上*，2013 年。

[24] Sandvine。加密的互联网流量：全球互联网现象的焦点。在 *Sandvine 关于全球互联网现象的报告中*，2016 年。

[25] J. Sherry，C。Lan，RA Popa 和 S. Ratnasamy。Blindbox：通过加密流量进行深度数据包检查。在 *ACM 数据通信特别兴趣小组会议（SIGCOMM），* 2015 年。

[26] T. Skybakmoen，J。Pathak，B。Venkateswaran，M。Spanbauer 和 B. Walder。违规检测系统比较报告。在《*NSS Labs 安全性价值地图》中，* 2016 年。

[27] B. Stricker。在加密流量中发现隐藏的威胁：对北美和 emea 的研究。在 *A10 和 Ponemon 研究所的报告中，* 2016 年。

[28] G. Vigna。UC Santa Barbara ictf 竞赛。https://ictf.cs.ucsb.edu/#/，2016 年。

[29] A. Yamada，Y。Miyake 和 K. Takemori。入侵检测，用于加密的 Web 访问。在 2007 年*高级信息网络和应用程序研讨会上*。

[30] Z. Zhou 和 T. Benson。迈向带有 qos2 的 https 和中间盒的安全游乐场。在 *ACM SIGCOMM 关于中间盒和网络功能虚拟化热点话题的研讨会上*，2015 年。

附录

A.安全协议

A.1 检测特性

证明。我们认为该方案的检测性能是成功的对手。根据第 2.3 节中给出的检测实验，这意味着存在一个关键字 w^*，使得：

1. SE 在 B 中发布了活板门 $T_{w^*} = F(w^*)$；
2. S 输出有效的密文 $c^* = (c_1^*, c_2^*, c_3^*, c_4^*)$；
3. 检测测试输出 1；
4. c^* 的解密返回 w^*。

特别是，第二点暗示：

$$c_3 = g_{i^2},$$
$$H(u^3) — c_3 = a,$$
$$c_i = g_{i,}$$

在哪里

$$= (c_3)^x,$$
$$(s_3, s_2) = G(s) >$$
$$吨^3 = C_2 ® 小号^3,$$
$$u = e(p^k RG, F(t_3)) 。$$

第三点暗示 c_4-H（u）= *a*，而 *u* = e（c_3，T_3）。然后我们得到

$$e(c_3, T_3) = e(pk\,RG, F(t^3)),$$
$$è(C_3, T_3) = E(G_{X\text{-}小号^2}, F(T^3)),$$
$$e(c^3_3, T_3) = e(g_{j^2}, F(t^3)^{x_i}) 。$$

由于 $T^3 = F(w^3)^{x_i}$ 并且由于在检测实验中，我们具有 Detect （T，R）= 0，这等效于 $t^3 = w^3$。然后

$$e(c_3, T^3) = e(g_{j^2}, T^3) 。$$

但是由于密文是正确的，因此等式 $c_3 = g_{i^2}$ 成立（请参见上文）。然后

$$e(c_3, T^3) = e(c_3, T^3),$$

这是无条件地不可行的。

我们最终评估误报的可能性，即与关键字 w 匹配的合法令牌 t，t = w。这意味着：

$$e(pk|^2_E, F(t)) = e(c_3, F(w)^x),$$
$$E(G_{X\text{-}小号^2}, F(t)) 的= E(G_{j^2}, F(w) 的^{x_i}),$$
$$E(G_{我}, F(t)) 的^{x_i\text{-}小号^2} = E(G_{我}, F(W))^{小号^2 \cdot X_i} 。$$

最终我们得到 F（w）= F（t），这是哈希函数 F 中的冲突，其发生概率很小。这证明了结论。□

A.2 流量不可分辨性

证明。我们用混合参数证明定理 2。我们为发送过程构造了一个模拟器 S，针对该实验的对手 *A* 拥有 1/2 的优势。为了获得这一点，我们希望 S 的输出是完全随机的。

我们可以假设 *A* 发送给质询者的消息与任何规则都不匹配。的确，如果它们匹配相同的规则，则不可区分性就没有意义；如果它们匹配不同的规则，则不可区分性将被轻易打破，尽管仍然存在单向性。

游戏 G_0。该游戏是原始游戏，这意味着 S 遵循发送过程。对手选择 T_0 和 T，挑战者加密其中之一。对手的观点是以下加密流量，其中 *b* 是挑战者选择的位。

$$\begin{matrix} Ç_{我,B} \\ Ç_{2,B} \\ c_{3,b} \end{matrix} \quad \begin{matrix} 壳^{[R]}{}_b 。 \\ 壳_{我} 。 \\ ^{s}1, B Ⓡ {}^{\dagger} b 。 \\ 克_{我} 。 \\ H(u_b) + a \end{matrix} \quad \begin{matrix} {}^{t_2, B} \\ \\ \\ \\ \bmod q 。 \end{matrix}$$

在整个游戏序列中，我们将 S_i 表示游戏 G_i 由 A 输出的位 *b* 为 b 的概率。我们有：

$$进阶?_{,} “A = 1\,2 \bullet Pr[Exp^{/r}{}_{A^{ind}} = 1] - 1| = |2 \bullet S_0 - 1| 。$$

我们强调 c_1 不依赖于 T_b，并且由于 r_b 是 Z_q 的随机元素而被完全随机化 。

游戏 G_1。然后，我们证明 c_2 也是 Z_q 中的随机元素（在随机预言模型中）。由于 c_2 的计算对应于一次填充，因此我们的工作是证明 s_1 是随机元素。实际上，s_1 是哈希函数 G 的输出，在此建模为随机预言。然后，对手的唯一可能性就是通过---------- 'r ^ 对随机预言的请求来获得 s_1。G 的输入是 pk_R，这意味着如果对手能够计算 $g^{/r}$，他可以恢复 s_1。----------。从 $C_{我}$=和 $PK_{小R} = G_4$，其精确地相当于实例 CDH 问题。

但是， 对抗 CDH 的对手 *B* 无法直接使用 A 赢得比赛。一种可能是使我们成为双线性图，但由于我们使用了不对称配对，因此他无法验证 A 对随机预言的请求中的哪一个确实是 CDH 挑战的答案。因此，对于所有针对 CDH 的对手 B ，用 g_0 表示 A 对随机预言 G 的请求数，我们显然有：

$$|Pr(S_0) — Pr(S_i)| < — Adv^{DH}_B(A) 。$$
$$q$$

游戏 G_2。现在我们关注 c_3。实际上，由于 $c_3 = g_{j^2}$，所以此步骤可以像以前一样对待，因为对手使用 c_3 来找到 b 的唯一方法是获得 s_2，从而向随机预言 G 发出正确的请求。显而易见，对于所有针对 CDH 的对手 B ，我们都有：

$$|Pr(S_i) — Pr(S_2)| < — Adv^{DH}_B(A) 。$$
$$q$$

游戏 G_3。我们最后将重点放在 c_4 上。在这个游戏中，我们替换 U = E（PK $_{小号}$Ë *，F（$T_{我}$）在 c）中的计算 $_4$ 由随机均匀值 u_3。我们认为，对手必须打破 GDDHE 问题的实例才能看到差异。因此，我们必须证明我们确实属于 GDDHE 实例。实际上，对手可以访问 $g^{p(r_b, s_2, x}$ 结合 c_1，c_2，$pk_{;}$，和 pk 的$_{SE}$。由于指数 r_b，s_2 和 *x* 被保密，因此所得多项式 P 具有最大 1 的单项式。A 还可以访问 $g^{q(“, x\”}$ 形式的元素，这里 a_i 是 F（t_i）的（未知）离散对数，A 可以为他选择的任何标记计算出那些令牌与发送给挑战者的令牌具有相同的类型。由于 a_i 未知，因此 *Q 的*每个单项式的阶数最多为 2，对应于形式为 $g_{j^{2\text{®}}}$ 元素 $^{'x_a b}$。但是，A 必须计算 G_i 中形式为 $g^{f(s_2, x'', a_b)}$ 的元素。指数 $f(s_2, x', a_b) = s_2 \bullet x' \bullet a_b$ 是阶数为 3 的单项式，显然它与 P 和 Q 线性独立。因此，对手的观点确实是一个实例的 GDDHE 假设。令 C 为

反对 GDDHE 假设的对手。如前所述，对于所有 C 并用 q_H 表示 A 对随机 Oracle 的请求数，我们有：

$$|\Pr(S_2) - \Pr(S_3)| < \frac{1}{q} \cdot \mathrm{Adv}_C^{\mathrm{DDHE}}(A)。$$

现在，S 的输出是一个完全随机的值，我们有 $S_3 = 1/2$。

因此，我们有：

$$\mathrm{Adv}_{\mathrm{SE}}^{\mathrm{IND\text{-}CCA}}(A) < \mathrm{Adv}_C^{\mathrm{DH}}(A) + \mathrm{Adv}_C^{\mathrm{DDHE}}(A)，$$

因此，对手的优势微不足道。□

A.3 规则不可区分性

证明。我们认为一个对手 $A = (A_f, A_g)$ 可以访问公共参数，包括 SE 公钥 $\mathrm{pk}_{\mathrm{SE}} = g_1$。对手 A_f 的第一部分在最小熵分布较高的情况下输出两个关键字 w_0 和 w_1，挑战者对输入 w_b 执行 RuleGen 以获得秘密比特 $b \in \{0,1\}$。这样的过程输出 $T_b = F(w_b)$。所得的 T_b 被提供给对手的第二部分 A_g。我们记住，A_f 和 A_g 不能相互通信，因为这将使它们轻而易举地打破规则不可区分性。

注意，T_b 的输出隐式地将输入 w_b 的随机预言子 F 的输出固定。

然后，根据规则不可区分性游戏，允许对手 A_g 使用其选择的任何关键字 w 创建其选择的任何加密流量，以执行方案 n 的发送过程。在每次执行时，对手应在输入关键字 w 时要求随机预言 F。如果要求 w_b，则 A_g 通过使用 e 的双线性特性来确定 b = 0 还是 1 变得容易，因为 $e(g_1, T_b) = e(\mathrm{pk}_{\mathrm{SE}}, F(w_b))$。但这仅以 $2^{-\mathrm{M}_\infty(A)}$ 的可忽略概率发生，其中 $\mathrm{M}_\infty(A) = \log A$，因为规则集具有较高的最小熵。否则，对手必须区分三元组（g_1，g_f，　　）和（g_1，g_f，h_f），且 h_0 和 h_1 未知（因为对应于 F 的未知输出），这显然是无条件地不可行的，因为对手没有足够的材料得出比随机猜测更好的结论。□