P2DPI：實用且深度保護隱私資料包檢查

金鐘吉1 , Seyit Camtepe2 , Joonsang Baek1 , 威利·蘇西洛1 , Josef Pieprzyk2,3和 Surya尼泊爾2

[1] 澳洲臥龍崗大學
[2] CSIRO Data61，澳大利亞
[3] 波蘭科學院計算機科學研究所，波蘭
Jongkil@uow.edu.au、Seyit.Camtepe@data61.csiro.au、Baek@uow.edu.au、Wsusilo@uow.edu.au, Josef.Pieprzyk@data61.csiro.au, Surya.Nepal@data61.csiro.au

抽象的。由於端對端流量加密的廣泛採用，加密網路流量幾乎每年翻一番

IPSec、TLS 和 SSH 等解決方案。儘管用戶有所有的好處
端對端加密提供的隱私，加密的網路流量
使入侵偵測系統（IDS)失明並導致偵測惡意
交通極為困難。由此產生的用戶隱私衝突
安全性需要深度資料包檢測 (DPI) 解決方案
透過加密的流量。迄今為止提出的解決方案的方法
仍然受到限制，因為它們在連接設定或檢測期間需要大量計算。例如Sherry推出的BlindBox

等人。 (SIGCOMM 2015) 可以在不損害使用者隱私的情況下檢查 TLS 加密流量，但由於以下原因，其使用受到限制
建立受檢查通道的時間明顯延遲。 PrivDPI，最近由 Ning 等人提出。 （ACM CCS 2019），提高了BlindBox的整體效率，並使檢查場景更加可行。

儘管有所改進，我們在本文中表明，用戶隱私
Ning 等人的 PrivDPI 可能會被規則產生器完全破壞
不涉及任何其他方，包括中間件。觀察到先前實現效率和安全的困難

在工作中，我們提出了一種用於加密流量的新 DPI 系統，名為「實用且隱私保護的深度資料包檢測（P2DPI）」。 P2DPI
享有與 BlindBox 提供的相同等級的安全性和隱私性。在
同時，P2DPI 提供快速設定和加密，且效能優於
PrivDPI。我們的結果得到了正式安全分析的支持。我們實施了 P2DPI 和類似的 PrivDPI，並進行了廣泛的研究
性能分析和比較的實驗。

關鍵字：深度包檢測；可搜尋加密；入侵偵測系統;滲漏系統

1 簡介

網路中間盒系統被廣泛採用來保護網路營運商和最終用戶。網路入侵偵測、入侵防禦和滲透系統是保護網路免受內部和外部攻擊的中間件解決方案的典型範例。中間盒系統執行深度資料包檢查（DPI)來評估資料包有效負載並檢查其對策略（例如，安全策略和資訊管理策略)的遵從性。一旦偵測到不合規情況，中間件就會觸發適當的操作來維護網路的安全。特別是，入侵偵測系統（IDS)和入侵防禦系統（IPS)透過監控網路流量來識別和防止惡意軟體入侵和安全策略違規等惡意嘗試。他們向安全審核員發出警報，並按照一組預先定義的規則執行自動保護操作。滲透系統可識別文件浮水印等特定簽名，並防止公司智慧財產權的洩漏。

儘管 DPI 作為組織網路的主要安全控制發揮著至關重要的作用，但對使用者隱私的需求極大地限制了 DPI 的使用。如今，網路上幾乎所有流量都是加密的。根據Google趨勢[1]，網路上排名前100的網站均支援HTTPS，它提供端對端加密。其中，96個網站預設使用HTTPS。此外，隨著對用戶隱私的擔憂增加，這是一個持續的趨勢。

DPI 的功能受到端對端加密（例如 TLS 協定)的嚴重破壞，因為加密使傳統的資料包檢查變得不可行。當端對端加密應用於流量時，只有兩端（例如客戶端和伺服器)可以共享內容。中間的任何系統都無法存取加密內容。

因此，中間盒技術（參見[21,18,10,22])自然地發展到檢查加密流量。這個方向的一個眾所周知的解決方案是使用「攔截代理」[10,17,9]，它的行為就像中間人（MITM)攻擊中的對手。此解決方案中斷客戶端和伺服器之間的身份驗證過程。因此，代理將端對端安全連線分為兩部分：使用者到代理（即代理充當客戶端的伺服器)和代理到伺服器（即代理充當客戶端的角色)到伺服器）。如果用戶端或伺服器使用具有完善的憑證鏈的數位憑證[10]，它可能無法正常運作。此外，這種方法會損害用戶隱私，因為客戶端和伺服器之間的所有流量都可以透過第三方代理存取。這些缺點成為採用這種方法的重要問題。

迪克森等人。 [6]提出了一種替代部署模型，其中所有中間盒功能均由最終用戶在主機上執行，以控制和監視其流量。這種基於主機的模型不會向最終使用者隱藏規則。然而，隱藏規則被認為是現代 IDS 的重要相容性要求。首先，根據 Paxson [20] 的說法，對最終使用者隱藏偵測規則使得攻擊者更難逃避使用者的 IDS。其次，根據[22]，偵測規則集很重要

IDS 供應商的智慧財產權。保持規則集在市場上更具競爭力可能是供應商的基本業務優先事項之一。

因此，從商業角度來看，對偵測規則保密非常重要。

由於攔截代理的明顯缺點，研究人員尋求另一種允許深度資料包檢查同時保護使用者隱私的方法。他們觀察到，如果加密允許搜索，那麼 DPI 和保護用戶隱私這兩個有爭議的屬性就可以實現 這就是眾所周知的可搜尋加密 [24,11,7]。可搜尋加密使中間盒系統能夠將加密資料包與加密搜尋關鍵字或令牌進行模式匹配[13,8,22,5]。請注意，在此解決方案中，所有搜尋操作都是在不解密加密資料包的情況下完成的。此外，搜尋令牌只能透過與使用者協商來產生。因此，只有有限數量的用戶認可的違規模式與加密流量進行匹配以進行檢測。此約束強制中間件執行僅由使用者允許的任務，並最大限度地減少使用者私有資料的潛在洩漏。

不幸的是，這樣的中間盒解決方案仍然不成熟並且存在其缺點。主要缺點是由於計算開銷過大而導致效能低。現有的解決方案需要緩慢的初始連接設定時間（約 97 秒)[22, 13] 或不切實際的檢查速度效能 [5]。

最近，Ning等人提出了PrivDPI。 [18]。它使用可重複使用的混淆顯著提高了 Sherry 等人的 BlindBox [22] 的效率。然而，與他們的聲稱不同，它沒有達到相同的隱私級別，因為客戶端和伺服器之間通訊的內容可能會被第三方洩露。

我們在後面的部分中透過提出損害「預處理協定」安全性的攻擊來證明 PrivDPI 不能保證與 BlindBox 相同的隱私等級。我們的攻擊基於這樣一個事實：PrivDPI 中的規則產生器不僅僅是創建規則來對加密流量執行簡單的詳盡訊息搜尋攻擊，這不適用於 Sherry 等人的 BlindBox。

因此，建立一個對於實際使用足夠高效的深度資料包檢測系統，但透過提供不允許濫用檢測能力的端對端加密來保護用戶隱私，成為一個基本問題。

1.1 我們的貢獻

我們提供了一種新協議，稱為「實用且保護隱私的深度資料包檢查（P2DPI）」。 P2DPI 旨在提供與 BlindBox [22] 相同等級的隱私。因此，它保證任何第三方，包括規則產生器和中間盒，都不能損害兩個最終用戶（例如客戶端和伺服器)的隱私。此外，中間件只能檢查與最終用戶協商的有限資訊。同時，從效率角度來看，P2DPI 也很實用，因為它在建立連線時不會有明顯的延遲。

早些時候，PrivDPI 旨在實現這些目標，將安全性和隱私性保持在與 BlindBox 相同的水平，但提高了效率。然而，我們觀察到 PrivDPI 容易受到窮舉訊息搜尋攻擊，這不適用於 BlindBox。因此，P2DPI 是第一個真正同時實現這些目標的協定。 P2DPI 使用密鑰同態偽隨機函數 (KH-PRF) 在用戶和中間件之間交換混淆的檢測規則，而不會洩露他們的混淆密鑰。採用KH-PRF為DPI系統帶來了顯著的改善。特別是，在保持相同安全等級的情況下，P2DPI 比 BlindBox 更有效率。與PrivDPI相比，P2DPI顯著提高了安全性和效率。

特別是，我們的論文提出了以下內容：

– PrivDPI 的安全分析 [18]：我們提供 PrivDPI 的安全分析。我們表明 PrivDPI 容易受到窮舉訊息搜尋攻擊。也就是說，在 PrivDPI 中，如果二進位訊息（0 或 1)被加密，則攻擊者可以輕鬆辨別哪一條訊息被加密。此漏洞使除兩個最終用戶之外的第三方能夠允許對訊息進行詳盡的搜尋。我們將在第 3 節中詳細分析 PrivDPI 協議。

– 提出P2DPI：我們提出P2DPI，它解決了PrivDPI的安全問題。 P2DPI保留了MBSE（中間盒可搜尋加密)對使用者隱私的安全性，並提供了正式的安全證明。因此，用戶的私人資料不能被任何一方洩露，包括中間件和規則產生器。因此，P2DPI 實現了與 BlindBox 相同的安全等級。而且，P2DPI 比 PrivDPI 更有效率。與 PrivDPI 相比，加密時間快 3.5 倍。例如，對於 1000 個訊息令牌，P2DPI 將加密時間從 55 毫秒減少到 16 毫秒。由於計算開銷減少，P2DPI 的設定時間也略有減少。 P2DPI的偵測時間與先前最佳的解決方案PrivDPI和BlindBox相當。

– 實證分析：我們提供P2DPI的實證分析。為了進行比較，我們實現了 P2DPI 和 PrivDPI 並比較了計算開銷。我們使用C語言來實作協議，部分使用python來模擬數位簽章。

我們強調 P2DPI 在安全性和安全性方面沒有任何缺點
當我們將其與 PrivDPI [18] 進行比較時，我們會看到效率方面的觀點。

本文的架構如下。第 2 節概述了我們的 P2DPI 以及相關威脅模型的描述。在第 4 節中，我們簡要描述了將在我們的協定中使用的概念和加密工具。第 5 節是本文的中心部分，討論了所提出的 P2DPI 協議的組件和結構。所提議協議的安全性和性能分別在第 6 節和第 7 節中進行分析。第 8 節概述了相關工作。我們在第 9 節結束我們的工作。

2 系統概述

2.1 系統架構

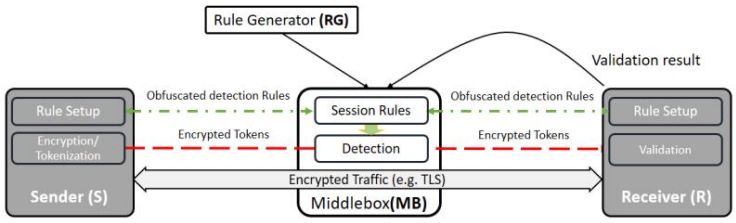

圖 1. P2DPI 系統。

P2DPI系統由四方組成 :規則產生器（RG） 、中間盒（MB） 、發送方（S)和接收方（R） ,如圖1所示。

RG是向MB提供檢測規則的一方。 RG與MB通訊僅用於部署和更新檢測規則 。通常 ,RG 被認為是誠實但好奇的。它可靠地產生檢測規則 ,但它可能會惡意行為或被惡意方脅迫以損害用戶的隱私。

MB 是入侵偵測系統或滲透系統 ,用於檢查從 S 傳送到 R 的加密流量。

MB 從 RG 接收（加密的)偵測規則。 MB 很誠實 ,但對 S 和 R 之間的交流感到好奇。

S 和 R 是使用類似 TLS 的協定相互安全通訊的最終使用者 。他們不想向 MB 或 RG 透露其通信的全部內容 。我們的解決方案假設至少其中一個（S 或 R)是誠實的。 S和R都對偵測規則感到好奇。 S 和 R 的角色可以互換 ,因為它們通常建立雙向通道 。我們假設S和R不能同時是惡意的。 （也就是說 ,S 或 R 至少是誠實的。)這是 IDS 和滲透系統前提的常見假設 。我們將在下一小節（第 2.2 節)中更詳細地討論這一點。

如圖 1 所示 ,發送方 (S) 和接收方 (R) 之間的 P2DPI 協定是透過與 MB 交換混淆格式的偵測規則來發起的。 MB和S/R成功共享偵測規則後 ,MB根據混淆後的規則計算會話規則 ,僅用於本次連線的偵測。 S 對流量進行令牌化並加密每個令牌 ,這表示為加密令牌 ,以使用會話規則中使用的相同加密將它們傳送到 R 。因此 ,MB 可以將偵測規則與封包負載進行匹配 ,同時兩者都被加密 。加密的令牌必須經過驗證

它們是否與透過加密通道（例如，HTTPS 服務的 TLS 加密)傳送的資料真正相符。這可以驗證S和R之一，它對MB是誠實的。請注意，我們假設圖 1 中的 R 是誠實的。

P2DPI協定中的MB、S和R之間的通訊是與傳輸層協定TLS（傳輸層安全)一起進行的。然而，檢查不一定在同一層中進行。

DPI 通常在網路層而不是傳輸層執行，因為它需要直接存取資料（有效負載)和每個資料包的標頭資訊[22]。從檢查的角度來看，P2DPI 與原始 DPI 沒有什麼不同，只是使用混淆格式的偵測規則對加密流量進行封包檢查。

2.2 威脅模型

MB (IDS) 的典型威脅：所有 IDS 解決方案都存在常見威脅 [13,5,22,18]。如果兩個端點都是惡意的，即 S 和 R 串通進行惡意活動，我們就無法在任何 IDS 系統中實施安全性。這是因為共謀端點可能會使用側通道 [20] 繞過任何控制，或在通訊系統中放置後門來加密流量以隱藏其模式。

請注意，在 BlindBox [22] 和 PrivDPI [18] 等先前的工作中，預設假設至少一個端點是誠實的。

我們注意到上述假設反映在目前 IDS 系統的操作中。如同[22]中所例證的，許多最近的滲透檢測系統都採取了「意外滲透」的預防措施，以消除使用者意外地以電子郵件附件的形式傳輸錯誤文件（例如公司的機密文件)的情況。（如果去掉這個案例，用戶就可以被認為是誠實的。)假設和練習的基礎是用戶遵守公司的政策，沒有任何繞過或欺騙IDS系統的意圖，而事件通常是由於以下原因發生的：正如Verizon 的報告中指出的那樣，他們的粗心行為[25]。從現在開始，我們假設接收者 R 是誠實的最終使用者。

BlindBox [22] 的作者指出，IDS 和滲透系統中的偵測規則必須對 S 和 R 隱藏，這也適用於我們的 P2DPI 系統。否則，S 和 R 可以操縱內容來逃避 IDS 和滲透系統使用的偵測規則。即使我們假設 S 和 R 之一是誠實的，這也是必要的。例如，引誘內部工作人員造訪其網站的惡意網站可能想了解偵測規則以繞過 IDS。

對使用者（S 和 R)的威脅：透過與 RG 協作，MB 通常透過檢查流量來提供入侵偵測和洩漏（請參閱 [22]）。我們假設MB和RG對S和R發送的數據內容感到好奇。

用戶 S 和 R 希望在同意深度資料包檢查的同時保留其通訊機密性。這裡，S和R透過限制偵測規則的數量來限制MB和RG的偵測能力。因此，MB 和

RG 執行深度資料包檢查，而不會損害 S 和 R 之間通訊的機密性（超出 S 和 R 同意的範圍）。

2.3 我們的技術概述

P2DPI 的關鍵挑戰是安全地隱藏各方之間的資訊。
MB和RG希望對S和R隱藏偵測規則。這些要求要求偵測規則必須以加密格式傳送給 S 和 R，但 S 和 R 需要在不知道偵測規則內容的情況下使用其金鑰對偵測規則進行加密。換句話說，S和R需要更新MB和RG給出的加密偵測規則的金鑰，而不需要解密它們。請注意，這是密碼學中眾所周知的做法，稱為「密鑰同態加密」。 ，

更準確地說，令 C = Enc(K1, M) 為密文，其中訊息 M 使用金鑰K1 加密。在金鑰同態加密中，更新者使用更新者擁有的金鑰K2將C更新為C = Enc(K1 · K2, M)，而無需解密C。解密且沒有金鑰K1，更新者無法取得M。

我們將使用密鑰同態加密來交換訊息，而不會洩露彼此的秘密。令KSR為S和R之間的共享金鑰，並令Mrule為偵測規則。我們協議中最大的挑戰是 MB 需要知道 Enc(KSR, Mrule)，而無需KSR或向 S 和 R透露Mrule。其從S和R更新回C dpi，可以透過計算Enc((KMB · KSR) · KMB來計算Enc(KSR, Mrule)。注意，金鑰同態加密演算法已經存在[3]。因此，我們可以透過採用現有技術之一來簡化DPI 技術。

= Enc(KMB · KSR, Mrule)。當MB收到C 解析度

$^{-1}$ ， Mrule)利用其KMB和密鑰同態知識。

此外，這些演算法被證明是安全的，因此它們在數學上保證滿足我們前面提到的要求。在第 4 節中，我們回顧了一個稍微概括的版本，「密鑰同態 PRF」（當然，它涵蓋了密鑰同態加密的定義）。

3 PrivDPI的安全分析

在本節中，我們表明 Ning 等人最近提出的 PrivDPI [18] 無法實現用戶隱私，這與他們聲稱「PrivDPI 保留與 BlindBox 相同的安全和隱私保證」相反。 （[18] 第 1.1 節。）

[4] 我們稍微濫用了「加密」一詞，因為規則實際上是混淆的而不是加密的。然而，我們在本小節中使用這個術語只是因為我們在高級視圖中解釋混淆是透過密鑰同態加密實現的。

如2.1節所述，DPI協定的典型設定涉及四個方（包括BlindBox、PrivDPI和我們的協定）：發送方（S）、接收方（R）、中間盒（MB)和規則產生器（RG）。 S和R是最終用戶，他們透過加密通道相互通訊。 MB 使用 RG 產生的偵測規則來檢查通道。特別地，S和R將透過使用相同的金鑰k加密來發送和接收資料令牌t。 MB將透過檢查t的加密令牌是否等同於r的加密偵測規則來確定加密流量中是否包含t。

根據 Sherry 等人的說法，他們的 BlindBox 協議不應該在 RG 和 MB 都受到對手控制（強制)的環境中使用 [22]。

然而，他們確實提到，即使 RG 和 MB 之一被對手脅迫，BlindBox 中的最終用戶也應該受到保護，免受攻擊，因為他們的流量受到比他們同意的更多規則的檢查。非正式地，它可以透過以下方式實現。

假設RG、MB和S/R之間商定了1000條規則。令 R 為包含這些混淆規則的集合，sig(R) 為由 RG 簽署的簽章。

MB 應該能夠只使用每個規則 r ∈ R 檢查加密流量。
(請注意，S 和 R 不知道 r 的內容。)這在 BlindBox 中可以透過以下步驟實現： 1) 驗證與 r 關聯的簽章後，S 向 MB 提供帶有硬編碼金鑰 k 的 AES 混淆； 2) MB可以在不知道k的情況下計算AESk(r)。

為了感受方案的安全性，請考慮以下兩種情況。

首先，假設 MB 受到損害，但 RG 沒有受到損害。由於S和R可以使用他們的簽章sig(R)和RG的公鑰來驗證偵測規則R是否來自RG。 MB不能在其上新增任何惡意偵測規則

自己的。

其次，假設 RG 受到損害，但 MB 沒有：RG 也不可能為新規則 r ∈ R/ 計算AESk(r)，因為 RG 不知道 k。

RG 無法自行產生它。因此，只要 MB 拒絕（秘密地)將 r 包含在 R 中，就可以獲得 S 和 R 的隱私。

現在，我們研究 Ning 等人的 PrivDPI [18] 的安全性。為了清楚說明，我們描述 PrivDPI 的「預處理」協議，該協議用於在 S、R 和 MB 之間設定混淆規則，如下所示。 （請注意，我們僅將[18]中使用者的概念 C（客戶端)和 S（伺服器)分別更改為 S 和 R。其他符號保持不變。）

PrivDPI的預處理協定[18]

輸入：MB 有輸入 ({si , Ri , sig(Ri)}i∈[N]) （來自 RG），其中Ri = g（這裡， ri是規則，α 和si是「致盲」 ri的 $^{\alpha ri+si}$ 值））； S 和 R 具有公共輸入 k。

1. S 計算KS = g 計算KR = g ^{k} （使用她的 k），並將KR傳送到 MB。類似地，R（使用她的 k)並將KR發送到 MB。
2. MB 檢查KS是否等於KR。如果不是，則停止並輸出 ⊥。否則，它將 ({Ri , Sig(Ri)}i∈[N])送到S。
3. S執行如下操作：

(a) 對於 i ∈ [N] ，使用 RG 所使用的簽章方案的公鑰檢查 Sig(Ri) 是否是Ri上的有效簽章。如果不是，則停止並輸出⊥。 kαri+ksi+k = g

(b) 計算Ki = (Ri · KS) ^{k} ^{2} 對於 i ∈ [N] 。最後，發送 {Ki}i∈[N]至 MB。

4. 對於 i ∈ [N] ，MB 驗證方程式 e(Ki , g) = e(Ri · KS, KS)是否成立。如果不是，則停止並輸出 ⊥ 。否則，對於 i ∈ [N] ，它計算kαri+k可重複使用混淆規則i = Ki/(KS) si以供將來使用。 = g ^{2}

RG的攻擊過程如下。 RG 只是觀察通訊狀態 ^{2} 上述協議中的MB和S之間的 kαri+ksi+k 。則由此得到KS和Ki = g ，這是可能的，因為這些參數未加密。然而，由於 RG 知道 α、 ri和si ，得到這個值的結果是RG可以計算附加會話規則2 k S = K ，換句話說，RG S g 可以為與預處理協定中處理的規則不同的新規則產生會話 規則，因此，PrivDPI 不保證與 BlindBox 提供的相同安全性。

，它可以獲得g ^{k} ^{2} 透過計算Ki/Kαri+si $_{S}$

對於任意規則 r $_{其}$ 透過計算S $_{其}$ ^{αr} kαr i+k = g ^{2}

此外，這種攻擊可以擴展到透過「窮舉訊息搜尋」攻擊來損害中間盒可搜尋加密（MBSE)安全性：RG 選擇代表低熵資料（例如溫度、日期甚至二進制數)的規則，並建立一個其本身的模糊規則。然後，它徹底搜尋加密流量以查找與規則相符的相應訊息（即其解密）。這種攻擊的後果是RG可以打破加密流量的兩個序列的不可區分性，從而損害MBSE安全性。（請注意，MBSE 安全性在第 6 節中正式定義。）

BlindBox 和 PrivDPI 處理規則的區別在於，在 BlindBox 中，RG 僅創建規則，而在 PrivDPI 中，RG 產生額外的值以及涉及 MB 和最終用戶的預處理協議中使用的規則。因此，只要MB拒絕處理RG未達成一致的規則，BlindBox中就實現了用戶隱私。

上述攻擊暗示PrivDPI的MBSE安全性分析有問題。請注意，MBSE 安全性對於誰可以成為對手沒有任何限制，也就是說，它保證用戶隱私免受任何對手的攻擊，包括 MB 和 RG。在 MBSE 安全性的攻擊遊戲中，對手會產生規則並（向挑戰者)查詢它們以獲得模糊的規則。這反映了BlindBox的真實情況，其中RG除了規則之外不產生任何其他值。因此，BlindBox [23]的安全證明適用於 RG 或 MB 可以成為對手的情況。然而，在PrivDPI的預處理協議中，RG產生其內部值，如α和si ，這些值在PrivDPI的MBSE安全證明中並未提供給對手[18]。換句話說，他們的證明並沒有解決 PrivDPI 中可能出現的真正隱私問題。

，

有人可能會說 RG 必須完全值得信任。然而，根據雪莉等人的說法，事實並非如此，因為它被認為是一些外部團體，例如非營利組織，並且可能受到對手（例如極權政府)的脅迫。 [22]。此外，我們預計，就 DPI 的一般功能而言，讓 RG 產生將由其他方（包括 MB)以完全信任的方式使用的參數是很麻煩的。與作者的說法相反，PrivDPI 沒有達到 BlindBox 提供的相同級別的安全性，因為 BlindBox 不允許這種攻擊，並且不信任 RG，而 PrivDPI 則信任 RG。

4 預賽

在本節中，我們概述了設計協議的主要技術，回顧了加密原語並總結了各種符號，這些符號將用於描述我們的協議。

4.1 關鍵同態PRF

Naor等人基於DDH（Decisional Diffie-Hellman)假設[16]的偽隨機函數（PRF），我們稱之為“密鑰同態PRF”，是P2DPI協議的重要組成部分。首先，我們回顧 PRF 的定義如下。

定義1（偽隨機函數（PRF)[3])一個有效可計算的函數F：K×X→Y，其中K是密鑰空間，X是輸入空間，Y是輸出空間，如果沒有有效的對手可以，則為PRF將 F 與僅透過黑盒存取 F 的真正隨機函數區分開來。

我們也回顧了 DDH 假設的定義如下。

定義2（DDH假設） 設Gp是素數階p的有限循環群，由g產生。 DDH 假設成立，如果給定 {g, gc1 , gc2 , T ∈ Gp}對於Zp中的均勻c1和c2，則沒有機率多項式時間 (PPT) 演算法可以確定 T 是 g ^{c1c2} 還是Gp中的隨機整數不可忽略的機率。

現在，我們回顧一下關鍵同態 PRF 的定義。

定義3（關鍵同態函數[16])假設k ∈ Zp是隨機均勻選擇的。另外，指定單向函數 H : X → Gp。

那麼，密鑰同態函數FKH定義如下：

$$FKH(k, x) := H(x)^{k}。$$

上述鍵同態函數具有以下性質，這些性質均在[3]中證明。

性質 1. (PRF)：如果 H 被建模為隨機預言，則密鑰同態函數FKH是 PRF。也就是說，PPT 對手 A 在區分FP RF和隨機函數的優勢小到可以忽略不計。

由於上述性質，我們將鍵同態函數稱為“key-
同態 PRF」。

性質 2.（關鍵同態）：對任意k1、 k2 ∈ Zp且任意 x ∈ X ，

$$FKH(k1, x) \cdot FKH(k2, x) = FKH(k1 + k2, x)。$$

性質 3.（交換性）：對任意k1、 k2 ∈ Zp且任意 x ∈ X ，

$$FKH(k1, x)^{k2} = \underbrace{FKH(k1, x) \cdot \cdots \cdot FKH(k1, x)}_{k2次}$$
$$= FKH(k1 + \underbrace{\ldots + k1}_{k2次}, x)$$
$$= FKH(k1 \cdot k2, x)。$$

類似地， $FKH(k2, x)^{k1} = FKH(k1 \cdot k2, x)$。

在整篇論文中，我們使用 $H(x, k) = (g\, H1(x)h)^k$，其中 g、h ∈ Gp、 k ∈ Zp且H1(·) 是可程式隨機預言 x ∈ X 關，機。注意，有些
鍵同態 PRF 的其他結構可以在 [16] 和 [3] 中找到。

4.2 符號

P2DPI 協定使用各種金鑰，每個金鑰都有不同的用途。
我們將它們總結如下。

– kMB：此金鑰由 RG 建立並與 MB 安全共用。它用於向 S 和 R 隱藏檢測規則（透過使它們與相同長度的隨機值無法區分）。請注意，公開此密鑰將允許 S 和 R 透過洩漏檢測規則來繞過檢查。 – kSR：此金鑰是 S 和 R 之間的共用金鑰，用於混淆它們的通訊。該密鑰的洩漏允許對手透過窮舉訊息搜尋攻擊潛在地讀取從 S 發送到 R 的封包。

該密鑰也用於更新MB發送的混淆檢測規則的密鑰。由於我們協定的密鑰同態屬性，它的值在更新期間不會透露給 MB。

– kSSL：這是用於 SSL/TLS 協定建立的安全通道的會話金鑰。我們假設該金鑰由 SSL/TLS 協定管理。

P2DPI中的偵測規則可以使用以下格式表示：

[5] 該版本的論文中，定義 $H(x, k) = (g$ [12] 修改為 $H(x, k) = (g\, H1(x)h)^k$ 由於先前選擇的可塑性，在先前的版本中。

– ri ：偵測規則（純文字格式），由RG產生。
– Ri ：RG 的混淆偵測規則。 ri透過密鑰 kMB 使用密鑰同態 PRF 混淆為Ri ， kMB被分配給每個 MB。
– sig(Ri) ：使用 RG 私鑰的混淆偵測規則Ri的簽章。可以使用 RG 的公鑰來驗證簽章。

– Ii ：中間混淆偵測規則。由於Ri使用密鑰同態 PRF 混淆，因此其密鑰可以從kMB更新為kMB · kSR，這裡kSR是 S 和 R 之間共享的混淆密鑰，在握手過程中達成協議。 Ii對應於ri，但使用更新的密鑰kMB · kSR 對其進行了混淆。

，

– Si ：會話規則，在資料包檢查期間使用。 Si對應但使用密鑰kSR（ S 和 R 之間的共享密鑰)進行混淆。
裡，

5 P2DPI協議

在本節中，我們將描述我們的 P2DPI 協定。

5.1 規則設定

在規則設定階段，RG為MB產生一個秘密金鑰kMB ，S和R產生並共用另一個金鑰kSR以準備深度資料包檢查。此外，也為 S 和 R 之間的 SSL/TLS 加密通道建立和共用kSSL 。

規則設定演算法（演算法 1)的主要想法是允許各方使用自己的金鑰來混淆偵測規則，以便保持規則隱藏安全性。這意味著RG和MB使用kMB混淆規則，並且用戶S和R另外使用密鑰同態屬性將已經混淆的規則Ri的密鑰從kMB更新為kMB · kSR 。現在，知道中間混淆規則Ii和金鑰kMB， MB從Ii中刪除kMB並產生會話規則Si ，這些規則僅在金鑰kSR下進行混淆。 MB 使用會話規則來檢查從 S 傳送到 R 的資料包。

5.2 加密

P2DPI 要求從 S 到 R 的流量 T 被解析為ti並進行混淆以隱藏內容並將其與所有 i ∈ [Nr]的會話規則Si進行匹配。 （請注意， Si在規則設定（演算法 1)中進行協商和混淆。)因此，加密階段涉及兩個過程：提取令牌和混淆令牌，如下所示

提取標記：我們的偵測演算法使用 BlindBox [22] 的標記化方法。

演算法1 :規則設定1. RG 生成g, h ∈ Gp ，

其中Gp是素數階p 的有限循環群。它還選擇與 MB 安全共享的隨機化密鑰kMB ∈ Zp ，並使用kMB將ri混淆為Ri ，如下所示： 對於 i ∈ [Nr] ，其中Nr是檢測規則的總數 ，計算

$$Ri = (g^{H1(裡)} H)^{千兆位元組}。$$

接著RG使用其私鑰將Ri簽章為sig(Ri) ，並將一組偵測規則{(Ri, sig(Ri)|i ∈ [Nr]}傳送到MB。

2. S和R向MB發送連線請求。
3. MB送{Ri, sig(Ri)|i ∈ [Nr]}到S和R。
4. S和R都使用RG的公鑰驗證sig(Ri) 。如果它們有效 ，則 S 或 R 產生kSR並使用 SSL/TLS 握手的金鑰共享協定與另一方共用 。否則，它們輸出 ⊥。
5. S和R都計算一組中間混淆規則

$$\{Ii = Ri^{kSR} (g^{H1(ri) = kMB \cdot kSR \cdot h)} | i \in [Nr]\}。$$

S 和 R 將{Ii|i ∈ [Nr]}回傳給 MB。

6. MB檢查S和R發送的{Ii|i ∈ [Nr]}是否相同 。如果沒有的話 ，它斷開S與R之間的連結。

– 基於 Windows 的標記化方法使用簡單的滑動視窗演算法從資料包中提取標記 。特別是 ，它按固定長度解析位元組流 。我們遵循 BlindBox 並使用 8 位元組視窗 。例如，「嚴格保密」一詞被解析為「嚴格」、「trictly」、「rictly c」、「ictly co」、「ctly con」、「tly conf」、「ly confi」等。

– 基於分隔符號的標記化使用標點符號或空格等分隔符號來解析基於文字的有效負載 。例如，「嚴格保密」可以使用空格解析為「嚴格」和「保密」。 “login.php”一詞可以解析為“login”和“php” ，其中“.”是分隔符號。

轉換令牌 :我們的加密演算法與 BlindBox [22] 的加密機制類似 ，只是將 AES 計算之一替換為金鑰同態加密 。我們令H2 : {0, 1} n2 × Gp → {0, 1} n3為可程式隨機預言 。特別地 ，透過運行下面給出的演算法 2 ，從流量 T 中提取的令牌ti被轉換為加密令牌Ti ：

請注意 ，計數器 c 用於隱藏加密令牌的模式 ，類似於 BlindBox 的 DPIEnc 。由於H1和H2的輸出是確定性的 ，因此如果存在相同的輸入 ，MB 可以學習加密資料的模式 。因此 ，使用計數器 c + i 來避免這個問題 。此外 ，由於隨機計數器 c 是共享的 ，並且每一步都加一 ，因此可以節省通訊開銷

演算法 2：加密1. 給定流量 T = {t1, ...,

tNt }，S 選擇隨機計數器 c 並為所有 i設定Ti = H2(c + i,(g H1(ti)h) kSR) ε [Nt]。

2. S 將 (c, {Ti|i ∈ [Nt])送到R（透過 MB）。
3. S 透過 MB 透過 HTTPS 連線將 T 傳送給 R。

head 並允許 MB在接收加密令牌之前從H2(c + i, Sj)準備匹配偵測規則。

5.3 檢測

我們的偵測演算法只需要對每個令牌進行Nr AES 計算，其中Nr是在規則設定演算法中協商的偵測規則的數量。因此，運行檢測演算法所需的計算與 BlindBox 和 PrivDPI 相同。下面（演算法3)描述檢測演算法。

演算法3：偵測1. MB從S接收(c, {Ti|i

∈ [Nt]})。
2. 對於所有 i，i ∈ [Nt]，對於所有 j ∈ [Nr]， MB 計算H2(c + i, Sj) 並將結果與Ti進行比較（如果 i = i）。

3. 找到匹配項後，MB 會發出警報。

5.4 驗證

惡意寄件者可能會嘗試透過建立不代表實際流量的加密令牌來逃避偵測。為了解決這個問題，我們假設 S 和 R 中至少有一個是誠實的。這是一個現實的假設，因為許多 Internet 應用程式都遵循客戶端-伺服器模型。我們的解決方案可用於保護誠實的 Web 伺服器免受惡意用戶端的侵害，或保護誠實的用戶端免受惡意 Web 伺服器的侵害。假設 R 是誠實的，驗證過程（演算法 4)描述如下。

6 安全分析

在本節中，我們對擬議的 P2DPI 協議進行安全分析。
我們首先回顧MBSE安全性的定義並定義規則隱藏安全性。
基於這些定義，我們證明了我們協議的隱私性和安全性。

演算法 4：驗證1. R 從 S 接收({ti|i ∈ [Nt]}, {Ti|i ∈ [Nt]}, c)。
2. R使用kSR將{ti|i ∈ [Nt]}標記化並加密為 {T i |i ∈ [Nt]} ， 3. 如果{Ti|i ∈ [Nt]} = {T i |i ∈ [Nt]]} ，R 警告 MB。

6.1 Middlebox可搜尋加密及其定義
安全

中間盒可搜尋加密（MBSE)[22, 24]是一種涉及RG 、MB 、S和R之間互動的協定。

中間盒可搜尋加密（MBSE)首先，我們定義一個（通用）
MBSE方案正式如下。

定義4（MBSE)中間盒可搜尋加密由五種演算法組成，分別為Setup 、RuleEnc 、Encrypt和Match：

– 設定 [1λ → (sk, pk)] 接受安全參數1λ並輸出加密
密鑰 sk 和公用參數 pk。
– RuleEnc [(sk, r, pk) → S)] 接受來自訊息空間 M 的 sk 和 r ，並且
輸出一個混淆的規則S。
– 加密 [(sk, T , pk) → (param, ET)] 接受 sk 和來自 M 的一組標記 T = {t1, ..., tn} ，其中n = poly(λ) 並輸出加密參數param和加密令牌 ET = {T1, ..., Tn}。

– Match [(param, ET , S) → I] 接受 param 、ET 和S。

正確性 對於任何足夠大的安全參數 λ ，對於任何多項式數 n = poly(λ) ，對於所有{t1, ..., tn} ∈ Mn ，對於每個規則 r ∈ M ，如果 S 是 RuleEnc 的輸出(sk, r, pk) 和(param, ET) 是Encrypt(sk, T , pk) 的輸出，其中sk 和pk 是Setup(λ) 的輸出。 Match(param, ET , S)輸出索引i的集合。微不足道。

使用者隱私 使用者隱私透過「MBSE 安全」保證 S 和 R 之間通訊的資料包的機密性。 （換句話說，MBSE安全保證了最終用戶的隱私。)這意味著MB可以根據RG產生的一組檢測規則來檢查從S發送到R（或R到S)的加密資料包，而其中的內容不符合檢測規則的資料包仍然保密。 MBSE 安全性的定義是基於不可區分性博弈，其中對手獲得了

其選擇的兩個相同長度的令牌序列的加密令牌，試圖區分它們。假設對手可以選擇任意數量的規則並獲得其加密，但這些規則不包括在用於加密查詢的令牌集中。 MBSE 安全性的正式定義如下。

定義 5（MBSE 安全）設 A 為有狀態 PPT 對手。考慮以下安全賽局ExpMBSE (1λ)：
A

– Init：挑戰者C以安全參數1λ執行Setup，產生規則加密金鑰sk和公用參數pk。它將 pk 發送給 A。

– 查詢：A 使用一組規則{r1, ..., rq}查詢 C。 C 使用 sk 呼叫 RuleEnc
並將{S1, ..., Sq}回傳給 A。
– 挑戰：A 從訊息空間 M 產生兩組標記T0 = {t0,1, ..., t0,n}和T1 = {t1,1, ..., t1,n}，使得 ∀i ∈ [q]， ri ∈
T / 0 ∪ T1。它將兩者都發送給C。 C 將 (param, ET) 回傳給 A。

– 猜測：A 輸出 β。如果 β = β，則 A 獲勝，遊戲輸出 1。

我們說 MBSE 是安全的，如果對於所有 PPT 狀態對手 A，則存在
一個可忽略的函數 negl 使得

$$\Pr[\mathrm{Exp}^{\mathrm{MBSE}}_{A}(1^{\lambda}) = 1] = \frac{1}{2} + \mathrm{negl}(\lambda)。$$

6.2 規則隱藏的定義及其安全性

規則隱藏可以防止S和R知道目前的偵測規則，因此無法繞過偵測機制。首先，我們正式定義規則隱藏方案如下：

定義6（規則隱藏方案） 規則隱藏方案由兩種演算法組成，即Setup 和Eval。

– Setup[$1^{\lambda} \rightarrow$ (sk, pp)] 接受Zp中的安全參數 1 金鑰 sk並發佈 Eval。它選擇隱藏規則
的公用參數 pp。
– Eval [(sk, r) → R)] 接受隱藏金鑰 sk 的規則和來自訊息空間 R 的規則 r。

我們的規則隱藏安全模型遵循典型的不可區分性遊戲：作為對手的最終用戶被允許在查詢階段知道一些隱藏規則，但無法獲得有關挑戰階段給出的目標隱藏規則的任何資訊。 （在我們的協議中，最終用戶無法向MB 查詢任何規則，但用戶可以觀察MB 的行為，並可能獲得有關（審查的）規則的一些資訊。我們的安全模型透過允許對手查詢來反映這一點。）遵循隱藏安全規則的正式定義。

定義 7（規則隱藏安全)設 A 為有狀態 PPT 對手。
考慮以下安全遊戲 ExpRH (1λ)，由四個步驟（初始化、查詢A挑戰、猜測)組成：

– Init：挑戰者C使用安全性參數1λ執行Setup以產生規則隱藏金鑰sk。

– 查詢：A 向 C 查詢規則ri，並將Ri混淆到 A。 ， 其中 i = [n − 1]。對於每個ri ，C 返回

– 挑戰：A 產生兩條規則r0,n和r1,n，使得r0,n和r1,n之前從未被查詢過，並將它們發送給 C。 C 將Rβ,n回傳給 A。

– 猜測：A 輸出 β ∈ {0, 1}，若 β = β ，A 獲勝，遊戲輸出 1。
否則，輸出 0。

我們說 P2DPI 擁有隱藏安全性的規則，如果對於所有 PPT 都有狀態 A，
存在一個可忽略的函數 negl 使得：

$$\Pr[\mathrm{Exp}^{RH}_{A}(1\lambda) = 1] = \frac{1}{2} + \mathrm{negl}(\lambda)。$$

6.3 MBSE 安全性證明

為了證明P2DPI協定的MBSE安全性（用戶隱私），我們從中提取一個MBSE方案，我們將其表示為MBSEP 2DP I，如下。

令H2 : {0, 1} n2 × Gp → {0, 1} n3為帶鹽的可編程隨機預言機，其將n2位作為鹽和群元素並輸出n3位。

MBSEP 2DP I由以下演算法決定：

– Setup[1λ → (kSR, pk)] 接受安全性參數1λ。它選擇Zp中的加密金鑰kSR。它將偽隨機函數FKH和隨機預言H2 的描述發佈為公共參數 pk。

– RuleEnc[(kSR, r, pk) → S] 接受(kSR, r, pk) 並計算會話規則 S
在哪裡

$$S = F_{KH}(k_{SR}, x_i)$$

– 加密[(kSR, {t1, ..., tn},pk) → (param, {T1, ..., Tn})] 接受kSR 、 pk 和一組令牌{t1, ..., tn }來自訊息空間 M for n = poly(λ) 並輸出 param = c0和加密令牌{T1, ..., Tn}，其中

$$T_i := H_2(c_0 + i, F_{KH}(k_{SR}, x_i))。$$

– 匹配 [(參數, ET , S) → I] 接受來自 param 的初始計數器c0、會話規則 S 和加密令牌 {Ti |i ∈ [n]} 並輸出索引集

$$I = \{i | S = T_i \quad , 1 \le i \le n\}。$$

我們證明 P2DPI 透過證明 MBSE 安全性來提供使用者隱私
上述MBSEP 2DP I方案的：

定理 1.（MBSEP 2DP I的 MBSE 安全性)對於密鑰同態 PRF FKH(kSR, xi) （其中H1被假設為隨機預言），MBSEP 2DP I方案是 MBSE 安全的，假設H2是隨機的神諭。

$$\Pr[\mathrm{Exp}^{\mathrm{MBSE}}_{\mathcal{A},\Pi_{\mathrm{MBSEP\,2DP\,I}}}(1^\lambda) = 1] \leq \frac{1}{2} + \mathrm{negl}(\lambda)$$

證明。可程式隨機預言H2和偽隨機函數FKH(kSR, xi)。
我們令Tt = {tt,i|tt,i ∈ T0 ∪ T1}。我們可以使用以下遊戲來證明MBSE的安全性：

– GameMBSE$_0$：這是定義 6.2 中定義的原始遊戲。
– GameMBSE$_1$：這與GameMBSE相同，除了對所有xi ∈ {r1, . 。 。 rq}∪ T0 ∪ T1， FKH(kSR, xi)替換為隨機值 RVi，並限制其保持相等（即，如果xi = xj (ε X)，則 RVi = RVj (ε Gp)）。

– GameMBSE$_2$：這與GameMBSE相同，只是對於所有xi ∈ T0∪ T1，H2(c0 + i, FKH(kSR, xi)) 替換為隨機值 RV，並限制它保留 i 和xi的相等性。

GameMBSE$_0$和GameMBSE$_1$是相同的，因為FKH(kSR, ·)的輸出是均勻分佈的並且保持相等。因為對手沒有kSR，所以它無法計算FKH(kSR, y) = (g H1(y)h) kSR for y /∈ {r1, 。 。 。 rq} ∪ T0 ∪ T1測試回傳是隨機的還是偽隨機的。

GameMBSE$_1$和GameMBSE$_2$是相同的，因為H2是隨機預言機。由於增量計數器的原因，如果t0,i = t1,j且 i = j（對於t0,i ∈ T0 和t1,j ∈ T0），對手會收到相同的隨機值。否則，它總是收到不同的隨機值。

6.4 規則隱藏安全性證明

為了證明P2DPI協定的規則隱藏安全性，我們從協定中提取規則隱藏方案，用RHP 2DP I表示，如下。

令H1 : {0, 1} n1 → Zp為可程式隨機預言機，以n1位元作為輸入並輸出Zp中的元素。 RHP 2DP I由以下演算法決定：

– Setup[1λ → (kMB, pp)] 接受安全性參數1λ。它在Zp中選擇一個混淆密鑰kMB。它將函數H1和生成器 g的描述發佈為公共參數 pp。

– Eval [(kMB, r, pp) → R] 從訊息空間接受kMB, pp 和規則 r
R. 它輸出混淆規則 R，其中

$$R := (g\ H1(r)h)\ kMB.$$

我們透過證明上述規則隱藏安全性簡化為 PRF 安全性來證明規則隱藏安全性適用於我們的 P2DPI 協定。

定理2.（P2DPI 的規則隱藏）假設DDH 假設成立。那麼，RHP 2DP I滿足隱藏安全性規則或

$$\Pr[\mathrm{Exp}^{\mathrm{RH}}_{A,\mathrm{P\,2DP\,I}}(1^\lambda) = 1] \leq \frac{1}{2} + \mathrm{negl}_{\mathrm{DDH}}(\lambda)。$$

證明。注意上面的 RHP 2DP I同態函數FKH(k, x) = (g H1(x)h) k中用於創建 R 的函數，我們可以 是密鑰，其模擬 FKH 的規則隱藏遊戲 ExpRH亦然 B ：每當A 查詢規則時ri處 ， 中 k 設定為kMB。因此，對於 A 使用 PRF 於查詢階段，B 將它們轉送到其預言機（可以是隨機函數或$\mathrm{prf}_{A,\mathrm{P\,2DP\,I}}$ ad- (1λ) 以取得答案並將其轉回 A。 ，B 隨機選擇b ∈ {0, 1 }並將其預言機轉發rb,n得到目標規則並將其轉發回A。 [ExpPRF (1λ) = 1] 。由於根據[16]和[3]在隨機預言模型中的DDH假設下FKH被證明是安全的PRF，因此我們有 Pr[ExpPRF (1λ) = 1] ≤ + negl$_{\mathrm{DDH}}$(λ)，得到下界定理

乙

乙 $\frac{1}{2}$

陳述。

該證明意味著對手（即最終用戶）無法區分混淆規則 R 和隨機值，即使它知道其他一些混淆規則

規則Ri = (g H1(ri)h) kKB for i = 1, . 。 。 , q ，其中 q 是規則查詢的數量。
因此，規則隱藏安全性成立。

7 績效評估

我們實作了 P2DPI 協定的發送方 (S)、接收方 (R)、規則產生器 (RG) 和中間盒 (MB)，並評估了它們的效能。我們的 P2DPI 效能分析是在基礎作業系統為 Ubuntu 18.04LTS、配備 Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz 3.60GHz 和 16.0 GB 記憶體的機器上進行的。我們使用C程式語言來實作P2DPI。對於大數和橢圓曲線運算，我們使用了 OpenSSL 函式庫 [19]。我們也使用 Python ECDSA 套件 [4] 進行數位簽章。

以下是每個 P2DPI 元件的實作描述。

S 和 R：我們使用 OpenSSL 函式庫實作了一個 SSL 用戶端作為發送方 (S)。我們沒有分析接收方的驗證演算法，因為它與發送方的加密演算法相同。

RG：在我們的 P2DPI 中，我們簡單地使用 Snort Emerging Threats 來建立檢測規則ri 。規則產生器 RG 使用 OpenSSL 函式庫根據 ri 計算Ri ，並使用 Python ECDSA 套件對Ri進行簽章以產生 sig(Ri)。

MB：我們也使用 OpenSSL 函式庫實作了中間盒 MB，以重新實作設定和偵測演算法所需的加密操作。

7.1 實驗環境

在本節中，我們將我們的效能結果與 PrivDPI 的效能結果進行比較。根據[18]，PrivDPI是使用Python實現的。然而，我們使用 C 程式語言來實作 P2DPI。因此，為了公平比較（不依賴底層程式語言的效能變化），我們使用 C 實作 PrivDPI。

我們的 P2DPI 需要哈希和橢圓曲線運算。我們使用OpenSSL的bench-mark指令來比較幾種候選演算法的效能。我們使用 AES 演算法實現了雜湊函數H1和H2，以對可編程隨機預言進行建模。更具體地說， H1和H2的構造如下：

$$H1(x) := AES128x(鹽), H2(y, h) := AES128h \bmod 2128\ (y),$$

其中 H1(·) 中的 Salt是實現而固定的 salt 值； h 是質數域 prime256v1 上的群元素的座標； x和y來自關鍵字和計數器的網域。

我們採用 AES，因為它速度很快，因為最現代的 CPU 為其提供了硬體加速器，例如 AES-NI 指令。 （我們利用 OpenSSL 庫中的 EVP CIPHER 物件來加速硬體加速器的加密。)我們使用 CBC 模式進行 AES 操作。

對於橢圓曲線運算（例如標量乘法），我們使用了 NIST P256 曲線，這是我們可以與 OpenSSL 函式庫一起使用的最快選項之一。

7.2 規則設定

在P2DPI中，RG、MB和S/R相互交互，以混淆的格式交換由RG創建的檢測規則，為S、R和MB之間的封包檢測做準備。我們從「emerging-trojan.rules」中選擇了 2,000 條規則，其中包含 5,565 個關鍵字，每個關鍵字都由 Snort 新興威脅規則的 8 個位元組大小進行標記。表1給出了每個規則的關鍵字數量。

一是RG允許的規則準備，只有在偵測規則有更新時才執行，大概幾天一次。另一種是MB和S/R之間的會話規則交換，每當發送方(S)和接收方(R)之間有新的連線請求時就執行會話規則交換。請注意，會話規則交換對於系統的可用性更為重要，因為它會影響在使用者（S 和 R)之間建立安全通訊通道的初始握手。我們分別測量規則準備和會話規則交換時間，以估計 RG 的計算開銷和會話建立的延遲。

P2DPI 在不犧牲效能的情況下顯示了上述安全性改進。規則設定所施加的計算開銷幾乎相同。與 PrivDPI 相比，P2DPI 甚至略有減少。我們比較

表 1. 規則大小

規則數量 關鍵	1 500 1000 1500 2000			
字數量 8 1366 2727 4156 5565				

表2中規則準備和會話規則交換的處理時間
3. 因為簽名和驗證所花費的時間壓倒了這些
計算並盲目區分 P2DPI 和 PrivDPI 之間的差異，我們還
測量了沒有簽名方案的開銷來呈現差異。這
沒有簽名方案估計的時間使用“(w/o Sign)”表示。

如表2所示，1000條規則包含2727個關鍵字，
PrivDPI 的準備時間大約需要 704 毫秒，簽章產生在
估計，以及沒有簽名產生的 52 毫秒，這些分別是
在我們的 P2DPI 中，略微減少至 698 毫秒和 46 毫秒。減少的原因是
規則生成機制的差異。與 P2DPI 相比，PrivDPI
需要一個額外的隨機化參數si ，對於每個關鍵字並儲存它
將其傳遞給MB。這一要求導致了這種差異，儘管它不是
重要的。

會話規則交換時間如表3所示。
P2DPI 的交換時間仍然略小於 PrivDPI，因為它需要
橢圓曲線上的計算量較少。 1000條規則，會話規則交換
P2DPI的處理過程需要2.84秒（有簽名驗證)和0.30秒
無需簽名驗證。在 PrivDPI 中，相同的過程需要 2.87 秒
0.33秒，比P2DPI稍慢。

表 2. RG 的規則產生時間（毫秒）

# 的規則	P2DPI	P2DPI 標誌）（無標誌）	PrivDPI（無	隱私DPI
1	1.631	0.378	1.915	0.662
500 351.780	23.073	354.765	26.058	
1000 698.084	45.921	704.071	51.908	
1500 1038.264	70.417	1043.622	75.776	
2000 1385.290	93.604	1393.663	101.977	

頻寬：與不允許檢查的普通 TLS 相比，P2DPI 消耗更多頻寬，因為它需要在

規則設定。我們提供規則設定所需的額外頻寬
表 4 中 MB 和 S（或 MB 和 R)之間的頻寬消耗。

與規則設定中的每個連接相比，需要少一個組元素
與 PrivDPI。

表 3. MB 和 S/R 之間的會話規則交換時間（毫秒）

# 的規則	P2DPI	P2DPI 標誌）（無標誌）	PrivDPI（無	隱私DPI
1	4.270	1.426	4.798	1.954
500 1419.184	148.895	1438.990	168.701	
1000 2835.729	296.774	2868.596	329.641	
1500 4317.805	449.506	4362.262	493.963	
2000 5741.694	601.072	5804.661	664.039	

表 4. 頻寬（KB）

規則數量 頻寬	1 500 1000 1500 2000				
頻寬（無符號）	1.2 206.9 413.1 628.0 840.4				
1.1 174.9 349.1 532.0 712.4					

7.3 加密

與 PrivDPI 相比，P2DPI 提供更快的加密，這是因為 P2DPI 對每個令牌進行加密的橢圓曲線操作較少，與規則設定演算法類似，P2DPI 只需要一次標量乘法，而 PrivDPI 需要一次標量乘法和一次點加法，我們的可能仍然比 BlindBox 的加密慢，因為 BlindBox 不需要任何橢圓曲線操作，然而，當我們將 P2DPI 的加密速度與 PrivDPI 的加密速度進行比較時，效能差距縮小了，更準確地說，P2DPI 的加密速度比 PrivDPI 快約 3.5 倍，效率的提升源自於P2DPI和PrivDPI的實現都使用了OpenSSL庫中的EC POINT MUL運算，即實現橢圓曲線上的標量乘法，在P2DPI 中，S和R將標量乘法應用於固定生成器g，而在PrivDPI 中，S 和R 將它們應用到任意點 g α，另一方面，在2 k · α · ti+k中，在PrivDPI 中，S 和R 將令牌tiI加密為g，以使MB 能夠將其與加密規則進行匹配（請參閱第3 節中PrivDPI 的預處理協議），由於S和R知道共享金鑰k和令牌ti，但不知道α，因此在連線建立後需要立即從MB接收公共參數A = g α，然後，他們將開始透過計算Ak · ti · g來加密所有令牌，準確地說，加密令牌需要進行標量乘法H1(ti) · kSR並添加 h kSR來計算生成器 g 上的 g 陷門子到 P2DPI 中的 gh kSR，而在PrivDPI 中則需要對任意點A = g 進行標量乘法，我們的微觀分析表明，前一種情況的計算速度比後一種情況快 5.5 倍，特別是，在生成器上的標量乘法大約需要 8 µs，而在著名的 OpenSSL 函式庫中，在任意點上的標量乘法需要 44 µs，所以，

–　　–

，

k^2 。

H1(ti) · kSR ·

α

事實證明，在固定產生器 g 上執行標量乘法是 P2DPI 加密演算法相對於 PrivDPI 的重大改進。

表 5. 加密時間（毫秒）

代幣數	P2DPI	PrivDPI	
1k 代幣	15.227	55.229	
2k 代幣	33.586	109.434	3k 代幣 46.945
163.826			
4k 代幣	62.480	218.338	
5k 代幣	76.122	272.535	

我們估計了一些流行網站[6] BBC（英國廣播公司，4.4 MB）、BOA（美國銀行，6.5 MB）和 NYT（紐約時報，10.5 MB）的 P2DPI 載入時間。如圖 2 所示，這些頁面的載入時間並不算太慢，1k 規則的載入時間為 1.1 秒到 2.3 秒，2k 規則的載入時間為 1.4 秒到 2.6 秒。此結果表明，與 PrivDPI 相比，P2DPI 提供了更快的載入時間，1k 規則的載入時間為 3.3 秒到 7.5 秒，2k 規則的載入時間為 3.6 秒到 7.8 秒。需要注意的是，簽章驗證時間不包括在載入時間中，因為只有當 S 和 R 收到更新或新規則時才需要驗證；此外，網頁平均由 8 個位元組進行標記。估算表明，與PrivDPI和BlindBox相比，P2DPI更加實用。我們在比較中排除了 BlindBox，因為根據 [18, 22]，它顯然只需要數十秒的連接設定。

圖 2. 熱門網站的載入時間

[6] 我們假設圖像檔案（例如*.jpg 和*.png）和視訊串流（例如*.mp4）不受監控。這裡所指的網頁大小不包括圖片檔案或串流影片的大小。

7.4 檢測

我們注意到，我們的檢測演算法在理論上沒有差異
P2DPI、BlindBox 和 PrivDPI。與 BlindBox 和 PrivDPI 一樣，P2DPI 提供
文獻中對加密令牌的最快檢測。檢測取決於
檢測規則的數量和令牌的數量。結果是
如表6所示。

表 6. P2DPI 檢測時間（毫秒）

數量 規則數量（關鍵字數量） 代幣數量	1 條規則 (8)	1000 條規則 (2727)
1 個代幣	0.037	0.807
1k 代幣	4.229	490.541

8 相關工作

BlindBox：雪莉等人。 [22] 介紹了第一個實用的 DPI 解決方案
加密的流量。雖然 BlindBox 能夠快速進行資料包檢查，但它有一個
缺點很少。 BlindBox 要求偵測規則透過秘密進行加密
金鑰，在發送者和接收者（即使用者/客戶端和
伺服器），但不能像我們的 P2DPI 那樣透露給中間件。實現
為此，BlindBox 使用不經意轉移 [15, 2] 和 Yao 的混淆方案 [26]，
這在計算上是昂貴的，並且在
連線已啟動。例如，BlindBox需要97秒來處理
3000條檢測規則，包含約10000個關鍵字。盲盒似乎
不適合需要頻繁重新連接的應用，例如
網頁瀏覽，為每個 HTTP 建立一個新的 TLS/SSL 連接
要求。

BlindIDS：來自 Canard 等人的 BlindIDS。 [5] 不需要任何額外的
計算以建立連線。它還提供市場合規性。 Blin-dIDS 隨機化檢測規則，即它使它們與

相同長度的隨機值。然而，BlindIDS 是基於可搜尋的
由雙線性映射（即配對）建構的加密方案。意思是
實際檢查時間比 BlindBox 慢很多數量級。為了
例如，使用一條規則對一個令牌的檢查次數會超過 34,000
慢幾倍。這使得 BlindIDS 不切實際。

外包中間盒：中間盒功能可以外包到雲端
[13, 8]，可能位於組織網絡之外。這種系統架構使 BlindBox 更加實用，因為網關之間的會話持續時間更長。

然而，這並沒有克服 BlindBox 對於需要頻繁重新連接的應用程式的根本缺陷。

其他方法：Yan 等人的方案使用隨機化演算法準備檢測規則[27]。他們的方案在設定階段不會出現長時間延遲。然而，它需要「管理伺服器」為發送者和接收者創建隨機化金鑰，並將隨機化規則傳遞到中間盒。

如果中間件和管理伺服器串通，該系統會洩漏有關使用者通訊的資訊。類似地，[9]中提出的稱為「SGX-box」的系統，有助於具有特殊硬體的中間盒即時取得使用者的會話金鑰。但這種方式的問題在於，中間盒必須始終在線以提供檢查服務，並且用戶和中間盒之間必須建立安全通道。

引入多上下文 TLS 協定（mcTLS)[17]來減輕對資料隱私的擔憂。它強制中間件獲得客戶端和伺服器的批准以進行檢查。然而，它仍然基於 MITM 方法，需要與中間件共用 TLS 會話金鑰。它還需要更多的計算和通訊開銷來進行檢查。

深度學習技術用於對加密流量進行分類[14]。但是，它僅使用其協定和一些其他特徵對流量進行分類。
無法用於檢查流量的內容。

檢測規則的表達能力：P2DPI採用精確的關鍵字匹配進行檢測。換句話說，它將關鍵字（即檢測規則)與流量進行匹配，如果匹配則發出警報。完整的 IDS 系統需要更具表現力的規則，例如正規表示式或腳本，而不僅僅是關鍵字。根據雪莉等人的說法。 [22]，單一關鍵字精確比對可以覆寫一般HTTP IDS應用程式所需政策的1.6% - 5%。如果使用多個關鍵字進行精確匹配，則覆蓋率會增加到 29 - 67%，具體取決於 HTTP IDS 應用程式。可以透過先應用精確關鍵字匹配，然後部分解密關鍵字匹配的流量，以將正規表示式或腳本中編寫的規則應用於流量來覆蓋正規表示式和腳本。然而，據我們所知，對於P2DPI、BlindBox和PrivDPI支援場景，還沒有有效的演算法來支援無需解密的表達規則。

9 結論

在本文中，我們介紹了一種新的深度資料包檢測協定 P2DPI。
我們討論了現有的深度包檢測解決方案在啟動加密通道[22]方面表現出不切實際的性能或安全性較弱，無法滿足深度包檢測（DPI)系統通常要求的安全性等級。值得注意的是，我們表明 PrivDPI [18] 的使用者隱私無法針對規則產生器得到保證。 P2DPI 解決了安全問題並達到了 DPI 協定所需的安全級別，最初在[22]中建議。此外，從效率角度來看，P2DPI 優於 PrivDPI（文獻中最高效的 DPI 協議）

據我們所知。 P2DPI 另外將封包加密時間提高了約 3.5 倍 。我們的結果得到了基於我們實施結果的正式安全分析和性能分析的支持。

參考

1.網路上的https加密。 https://transparencyreport.google.com/https/overview?hl=en。
訪問時間 :2019 年 7 月 28 日。

2. 吉拉德·阿沙羅夫 、耶胡達·林德爾 、托馬斯·施奈德和邁克爾·佐納 。更有效率的不經意傳輸和擴展可實現更快的安全運算。 ACM CCS 2013 ,德國柏林 ,2013 年 11 月 4-8 日 ,第 535-548 頁 。美國CM ,2013。

3. 丹 ·博內 (Dan Boneh) 、凱文 ·路易斯 (Kevin Lewi) 、哈特 ·威廉 ·蒙哥馬利 (Hart William Montgomery) 與阿南斯 ·拉古納森 (Ananth Raghunathan)。
關鍵同態 prf 及其應用。 CRYPTO 2013 ,美國加州聖塔芭芭拉 ,2013 年 8 月 18-22 日 。施普林格 ,2013。

4.布萊恩華納 。純 python ecdsa ,2020。

5. S´ebastien Canard 、A da Diop 、Nizar Kheir 、Marie Paindavoine 和 Mohamed Sabt。
Blindids :基於加密流量的符合市場要求且保護隱私的入侵偵測系統。 ACM AsiaCCS 2017 ,阿拉伯聯合大公國阿布達比 ,2017 年 4 月 2-6 日 ,第 561-574 頁 。美國CM ,2017。

6. 科林·迪克森 、哈迪普·烏帕爾 、維耶科斯拉夫·布拉伊科維奇 、戴恩·布蘭登 、托馬斯· E·安德森和阿爾文德·克里希那穆西。 ETTM :可擴充的容錯網路管理員 。摘自David G. Andersen 和Sylvia Ratnasamy ,編輯 ,第八屆USENIX 網路系統設計與實作研討會論文集 ,NSDI 2011 ,美國麻薩諸塞州波士頓 ,2011 年3 月30 日至4 月1 日。 。

7. 湯瑪斯 ·福爾和帕斯卡 ·帕利爾 。可解密可搜尋加密。 ProvSec 2007 ,澳洲臥龍崗 ,2007 年 11 月 1-2 日 ,會議記錄 ,LNCS 第 4784 卷 ,第 228-236 頁 。施普林格 ,2007。

8. 郭宇 、王聰 、賈曉華 。在外包中間盒中啟用安全和動態的深度資料包檢查。 SCCAsiaCCS 2018 ,韓國仁川 , 2018 年 6 月 4 日至 8 日 ,第 49-55 頁 。美國CM ,2018。

9. Juhyeng Han 、Seong Min Kim 、Jaehung Ha 和 Dongsu Han。 Sgx-box :使用安全中間盒模組實現加密流量的可見性。 APNet 2017 ,中國香港 ,2017 年 8 月 3-4 日 ,第 99-105 頁 。美國CM ,2017。

10. 黃林雄 、亞歷克斯·賴斯 、埃爾林·埃林森和科林·傑克遜 。分析野外偽造的 SSL 憑證。 IEEE S&P 2014 ,美國加州伯克利 ,2014 年 5 月 18-21 日 ,第 83-97 頁。 IEEE 電腦協會 ,2014 年。

11. 塞尼·卡馬拉 、查拉蘭波斯·帕帕曼蘇和湯姆·羅德爾 。動態可搜尋對稱加密 。載於 Ting Yu 、George Danezis 和 Virgil D. Gligor ,編輯 ,ACM CCS 2012 ,美國北卡羅來納州羅利 ,2012 年 10 月 16-18 日 ,第 965-976 頁 。美國CM ,2012。

12. Jongkil Kim 、Seyit Camtepe 、Joonsang Baek 、Willy Susilo 、Josef Pieprzyk 和 Surya尼泊爾。 P2DPI：實用且保護隱私的深度資料包檢查 。載於Jiannong Cao 、Man Ho Au 、Zhiqiang Lin 和 Moti Yung ,編輯 , ASIA CCS 21 :ACM 亞洲計算機和通訊安全會議 ,虛擬活動 ,香港 ,2021 年 6 月 7-11 日 ,第 135-146 頁。美國CM ,2021。

13. Chang Lan 、Justine Sherry 、Raluca Ada Popa 、Sylvia Ratnasamy 和Zhi Liu。
著手 :將中間盒安全地外包到雲端 。卡特琳娜· J·阿吉拉基

和 Rebecca Isaacs，編輯，USENIX NSDI 2016，美國加州聖克拉拉，2016 年 3 月 16-18 日，第 255-273 頁。USENIX 協會，2016 年。

14. 穆罕默德·洛特福拉希(Mohammad Lotfollahi)、邁赫迪·賈法裡·西亞沃沙尼(Mahdi Jafari Siavshani)、拉明·希拉利·侯賽因·扎德(Ramin Shirali Hossein Zade)和穆罕默德薩德·薩伯里安(Mohammdsadegh Saberian)。深度資料包：一種使用深度學習進行加密流量分類的新方法。軟體計算，24(3)：1999–2012，2020。

15.莫尼·納爾和班尼·平卡斯。透過自適應查詢進行不經意的轉移。載於 Michael J. Wiener，編輯，CRYPTO1999，美國加州聖塔芭芭拉，1999 年 8 月 15-19 日，Proceedings，LNCS 第 1666 卷，第 573-590 頁。施普林格，1999。

16. 莫尼·納爾、班尼·平卡斯和奧馬爾·萊因戈爾德。分佈式偽隨機函數和 kdc。 EUROCRYPT 1999，布拉格，捷克共和國，1999 年 5 月 2-6 日，會議記錄，LNCS 第 1592 卷，第 327-346 頁。施普林格，1999。

17. David Naylor、Kyle Schomp、Matteo Varvello、Ilias Leontiadis、Jeremy Blackburn、Diego R. L´opez、Konstantina Papagiannaki、Pablo Rodr´ guez Rodr´ guez 和 Peter Steenkiste。多重上下文 TLS (mctls)：在 TLS 中啟用安全網路內功能。摘自 Steve Uhlig、Olaf Maennel、Brad Karp 和 Jitendra Padhye，編輯，2015 年 ACM 數據通信特別興趣小組會議記錄，SIGCOMM 2015，英國倫敦，2015 年 8 月 17-21 日，第 199 頁212.美國CM，2015。

18.Jianting Ning、Geong Sen Poh、Jia-Ch ng Loh、Jason Chia 和 Ee-Chien Chang。Privdpi：具有可重複使用模糊規則的隱私保護加密流量檢查。 Lorenzo Cavallaro、Johannes Kinder、XiaoFeng Wang 和 Jonathan Katz，編輯，2019 ACMCCS 2019，英國倫敦，2019 年 11 月 11-15 日，第 1657-1670 頁。美國CM，2019。

19.OpenSSL 軟體基金會。 Openssl，2020。

20.弗恩·帕克森。 Bro：即時偵測網路入侵者的系統。電腦網絡，31(23-24)：2435–2463，1999。

21. 安東尼奧·誇德拉·桑切斯和哈維爾·阿拉西爾。一種新穎的盲流量分析技術，用於偵測 WhatsApp VoIP 通話。國際。網路管理雜誌，27（2），2017。

22. Justine Sherry、Chang Lan、Raluca Ada Popa 和 Sylvia Ratnasamy。 Blindbox：對加密流量進行深度資料包檢查。 ACM SIGCOMM 2015，英國倫敦，2015 年 8 月 17-21 日，第 213-226 頁。美國CM，2015。

23. Justine Sherry、Chang Lan、Raluca Ada Popa 和 Sylvia Ratnasamy。盲盒：對加密流量進行深度資料包檢查。 IACR 加密。 ePrint Arch.，2015：264，2015。

24. 黎明曉東宋、大衛·A·瓦格納和阿德里安·佩里格。搜尋加密資料的實用技術。 IEEE S&P 2000，美國加州伯克利，2000 年 5 月 14-17 日，第 44-55 頁。 IEEE 電腦協會，2000 年。

25.版本。內部威脅報告。 https://enterprise.verizon.com/resources/reports/insider-威脅報告.pdf，2020 年。

26. 姚期智。如何產生和交換秘密（擴展摘要）。FOCS，加拿大多倫多，1986 年 10 月 27-29 日，第 162-167 頁。 IEEE 電腦協會，1986 年。

27.袁興良、王新宇、林建雄、王聰。外包中間盒中的隱私保護深度資料包檢查。 IEEE INFOCOM 2016，美國加州舊金山，2016 年 4 月 10-14 日，第 1-9 頁。 IEEE，2016。