基於雲端環境下智慧製造系統之安全通道實作分析

江忠晏[1] 羅乃維[2]
國立臺灣科技大學[1, 2]
b10809024@mail.ntust.edu.tw[1]
nwlo@cs.ntust.edu.tw[2]

摘　要

雲端及智慧製造的興起促使許多中小製造業從原先的本地部署（On-Premises）架構，轉變為雲端架構（Cloud），舉凡基礎設施即服務（Infrastructure as a Service）、平台即服務（Platform as a Service）或軟體即服務（Software as a Service）皆是雲端架構的範疇。然而即便雲端架構具備降低硬體、IT 及部署時間成本、增加靈活性等優點，但勢必也會面臨雲端服務供應商安全、網路傳輸安全等資通安全議題。故本研究分析中小製造業從本地部署架構轉型至雲端架構之中的安全架構建置及安全通道傳輸性能。

本研究基於 IaaS 架構，使用 IPsec 建置工廠內部路由器至雲端服務站到站（Site-to-Site）安全通道，其中安全通道採用的對稱式加密演算法為 AES128，雜湊演算法為 SHA1。最後使用 iPerf3 及 Netperf 分別地分析站對站（Site-to-Site）及端對端（End-to-End）網路吞吐量（Throughput）及延遲（Latency），並透過開源軟體 Wireshark 進行封包分析，以驗證安全通道安全性、效能及實用性。

關鍵詞：安全通道、雲端架構、智慧製造、網路效能

1. 前言與研究動機

2020 年全球深受新冠肺炎影響，許多產業開始採取遠距辦公，也有許多產業開始進行數位轉型，台灣的中小製造業也不例外，所幸經濟部從 2018 年起籌組智慧製造輔導團，協助我國中小企業導入智慧製造，提前做好準備。因此現今中小製造業實施智慧製造更是個趨勢，其中雲端技術是不可或缺的一環。

從 2019 年臺灣企業雲端大調查結果中能發現，臺灣的一般製造業上雲比例達到近 50%，其中最常上雲的應用為 24 小時服務或非關鍵性任務的應用。這項統計數據也意味著目前越來越多製造業接觸到雲端服務，將工廠內可程式化邏輯控制器（Programmable Logic Controller）的資料透過單晶片（Microcontroller Unit）上傳雲端。

從傳統的本地部署（On-Premises）架構如圖 1-1，即將資訊科技（IT）的系統服務（例伺服器、資料庫等）及操作技術（OT）置於工廠本地端，轉向將雲端架構（Cloud）如圖 1-2，即將資訊科技的系統服務部署於雲端伺服器，以利工廠管理階層在外能透過手機經由網際網路監測數據，因此雲端技術亦是架構中重要的一部分[1]，對於普遍企業使用雲端服務的優點不外乎為以下幾點，降低 IT 預算及時間成本、具有服務調節及擴展性、易於系統維護、易於管理控制等。

不過一體兩面，若將資料上傳雲端就容易延伸出網路傳輸安全、資料傳輸完整性、資料外洩等資通安全問題。

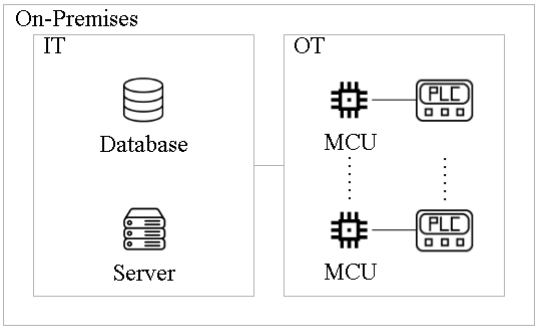

圖 1-1 本地部署架構

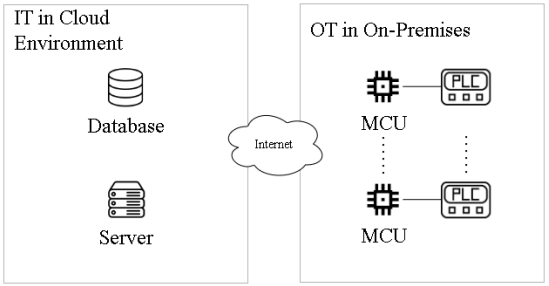

圖 1-2 雲端架構

由於網路傳輸安全面向過於廣闊，故本研究僅針對於工廠內部路由器至雲端服務伺服器之間傳輸性能及安全。因此，本研究在研究方法及步驟中，首先探討如何建立工業物聯網、雲端服務伺服器及手機應用程式，以模擬中小製造業上雲環境，再來研究如何利用工廠內部現有的路由器，建立至雲端服務的安全通道（IPsec），最後分析安全通道吞吐率（Throughput）、延遲（Latency），及驗證安全通道安全性。

2. 文獻探討及回顧

在文獻探討及回顧中，本研究探討物聯網安全及架構、安全通道建構技術及 IPsec 技術三個部分。

2.1 物聯網安全及架構

Rao 等人及 Kirupakar 和 Shalinie 皆提到隨著物聯網蓬勃發展，資安疑慮也是需要重視的[2, 3]。Kirupakar 和 Shalinie 於[3]所做的研究為在於工業物聯網環境下，由於現有的邊緣節點裝置（Gateway）為低功率，有著受限的 CPU 及記憶體性能，因此實作輕量化的入侵檢測系統（Intrusion Detection System）。當外部進行 DDOS 攻擊時，偵測任一邊緣節點的 CPU 使用率、記憶體使用率等相關重要指標，若是相鄰節點不在相對數值範圍內的話便進行入侵警報。

以系統軟體架構的方面，Sun 等人於[1]提出了一個基於企業服務匯流排（Enterprise Service Bus）所延伸出的雲端服務匯流排（Cloud Service Bus）SaaS 平台，其主要目的為整合生產商用軟體於他們所提出的架構平台上，例如將 ERP 及 CRM 等軟體部署在平台上，不僅減少中小企業購入、管理及維護等 IT 預算，中小企業更能夠透過網際網路，進行存取部署於雲端生產商用軟體。

2.2 安全通道建構技術

在安全通道建構技術中，本研究分別探討虛擬私人網路及穿隧協議兩部分。

2.2.1 虛擬私人網路

使用虛擬私人網路（Virtual Private Network）不僅能以廣域網域的便宜價錢，享受到專屬線路的安全，對於建置及維護更為彈性。Goethals 等人的研究[4]指出當前資料的安全是必須重視的，因此透過 VPN 傳輸能確保資料的安全性。

2.2.2 穿隧協議

目前有以下四種主流的穿隧協議（Tunneling Protocol），分別為通用路由封裝（Generic Routing Encapsulation）、端對端隧道協定（Point to Point Tunneling Protocol）、第二層穿隧協議（Layer 2 Tunneling Protocol）、網際網路安全協定（Internet Protocol Security）。

Jahan 等人的研究結論[5]提到，GRE 適合於時間敏感的應用程式；IPsec 則適合於安全性敏感的應用程式；PPTP、L2TP 並無加密機制，需搭配其他協議才可進行加密，例 L2TP over IPsec，且 L2TP 需設定使用者端軟體，架設較為麻煩。目前企業所使用的 CPE-based VPN 中，鑒於其安全性，故以 IPsec VPN 為主流。

2.3 IPsec 機制

在 IPsec 機制中，本研究探討封裝模式、安全協議、安全關聯、網際網路金鑰交換及 XFRM 框架五個部分。

2.3.1 封裝模式

IPsec 提供了兩種封裝模式（Encapsulation Mode）[6]可作使用，分別為傳輸模式（Transport Mode）及隧道模式（Tunnel Mode）。傳輸模式建立在兩台主機上，兩邊的主機都要架設 IPsec 協議，而隧道模式則是透過雙方閘道器或路由器進行 IPsec 協議。

2.3.2 安全協議

IPsec 具有兩種安全協議（Security Protocol）[7,8]，分別為 AH（Authentication Header）及 ESP（Encapsulating Security Payload）。AH 及 ESP 都具備資料完整性、資料來源驗證，然而 AH 並不支持資料加密，故此篇著重於 ESP 封裝協議[8]，而其封包格式如圖 2-1，Security Parameter Index 為安全關聯索引值[6]，Sequence Number 目的為反重播攻擊，Payload 為加密後的網路層封包（Layer 3 Packet），Padding 為對齊資料長度成 32 Bytes 整數倍，Padding Length 為填補資料的長度，Next Header 為 Payload 的協定，Authentication Data 為封包範圍的雜湊值。

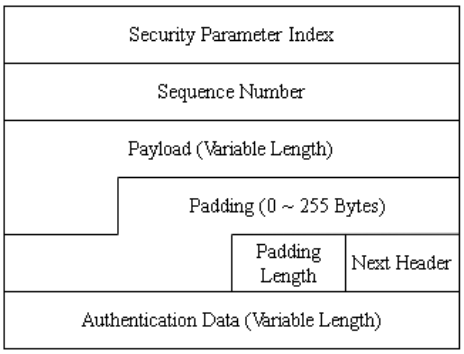

圖 2-1 ESP 封裝協議封包格式

2.3.3 安全關聯

IPsec 中最重要的便是安全關聯（Security Association）[6]，因為它定義雙方要如何協商，並儲存了雙方各種參數，例如要使用何種加密演算法？要使用何種封裝協議？SA 的生命週期多久？而這些安全關聯會放在主機或路由器的安全關聯資料庫 SAD（Security Association Database）中。

2.3.4 網際網路金鑰交換

網際網路金鑰交換（Internet Key Exchange）[9]能夠使雙方自主協調，產生 ISAKMP SA 及 IPsec SA，ISAKMP SA 是用以確保私鑰安全，其中參數

包括認證方式、加密演算法、雜湊函式等，而 IPsec SA 如 3.3 安全關聯所述為進行通訊時所規範參數。其中，IKE 又分作為 Phase 1 跟 Phase 2，Phase 1 主要目的為雙方互相認證，Phase 2 是基於 ISAKMP SA，可以執行多次，產生出多組 IPsec SA，假設 IPsec SA 的生命週期為 3600 秒，代表著每一個小時就會更新一筆 IPsec SA，但這一切都是由 ISAKMP SA 所確保其安全。

2.3.5 XFRM 框架

XFRM 為實作 IPsec 的框架，在傳送封包前會先經過 xfrm4_lookup 的方法，其作用為查詢是否有對應的 Outbound SA，若是符合 Outbound SA，則進入 XFRM 框架內根據模式及協議進行處理，最後執行 ip_output，以完成封包發送[10]。

在接收封包後會先偵測封裝類型，如果封裝類型為 AH、ESP，則會進入 XFRM 框架內查詢對應的 Inbound SA，再透過安全關聯中的模式及協議進行處理，處理完後將原始封包擷取出來重回到 ip_local_deliver，再經過 udp_rcv 或 tcp_rcv，最終將封包傳送至 socket[10]。

3. 研究方法及步驟

本研究著眼於中小製造業從本地部署架構轉變為雲端架構如圖 3-1，所將遭遇到的傳輸安全問題。

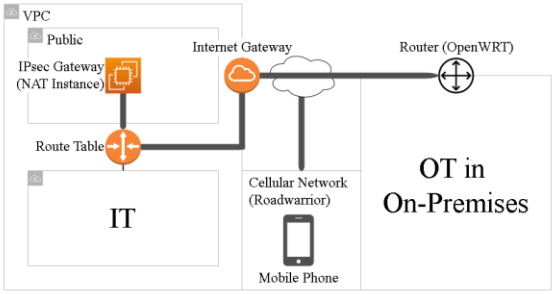

圖 3-1 中小製造業雲端架構模擬設計

3.1 上雲架構模擬設計

在上雲架構模擬設計中，分成工業物聯網架構流程、雲端服務架構流程兩部分進行。

3.1.1 工業物聯網架構流程

本研究採用 MediaTek LinkIt 7688 開發版以模擬工業控制器，以及採用 TP-Link Archer C7 AC1750 作為工廠端路由器，網路環境設定如表 3-1。之所以使用 LinkIt 7688 開發版及 Archer C7 AC1750，是因為其環境皆基於 OpenWrt 作業系統[11]，預設安裝了許多物聯網裝置開發常用的套件，包含多種程式語言的支援，例如 Python、Node.js 及 C 語言[12]。

因此，本研究在 LinkIt 7688 開發版上，撰寫簡單的定時獲取系統時間程式，並上傳資料到雲端服務中的資料存取伺服器，以模擬工業物聯網中將資料定時上傳模式。此外，在路由器上實作安全通道並分析其吞吐率、延遲及安全性。

表 3-1 路由器網路環境設定

TP-Link Archer C7 AC1750 網路設定	
Protocol	PPPoE
Public IP	106.104.X.X
LAN IP	192.168.2.1
Subnet IPv4 CIDR	192.168.2.0/24
Operating System	OpenWRT 21.04

3.1.2 雲端服務架構流程

在「雲端服務架構」中，本研究採用 Amazon Web Services 以作為雲端服務架構的模擬。首先建立 VPC（Virtual Private Cloud）雲端環境，VPC 的 CIDR 為 172.16.0.0/16。且在 VPC 中配置 Internet Gateway 以連通內外雙向網路，再分別於建立兩個 Subnet，環境設置如表 3-2。在 Public Subnet 中建立一 EC2 Instance 作為 NAT Instance，在 IT Subnet 中建立 EC2 Instance 作為資料存取伺服器（Server Instance），EC2 Instance 環境設置如表 3-3。

最後透過 Route Table 將外部流量導至 NAT Instance 上，經由 NAT Instance 轉發給其他子網路，或是子網路流量經由 NAT Instance 導至 Internet Gateway，Route Table 設置如表 3-4。此外，使用 Docker、Flask 及 MongoDB 建立具有容器化的資料存取伺服器。

表 3-2 Subnet 環境設置

	IPv4 CIDR
Public Subnet	172.16.1.0/24
IT Subnet	172.16.2.0/24

表 3-3 EC2 Instance 環境設置

	IPv4	Elastic IP	Type
NAT Instance	172.16.1.52	3.114.X.X	T3.medium
Server Instance	172.16.2.116	-	T2.micro

表 3-4 Route Table 環境設置

Public Subnet		
Destination	Target	Environment
172.16.0.0/16	Local	VPC
0.0.0.0/0	Internet Gateway	Extranet
IT Subnet		
Destination	Target	Environment
172.16.0.0/16	local	VPC
0.0.0.0/0	NAT Instance	To NAT

3.2 安全通道建置及分析

本研究透過 strongSwan[13]建置工廠本地端

至雲端服務端 site-to-site 安全通道（IPsec），以及雲端服務端至行動裝置端的 Roadwarrior 安全通道（IPsec），其簡化網路拓撲如圖 3-2 所示。

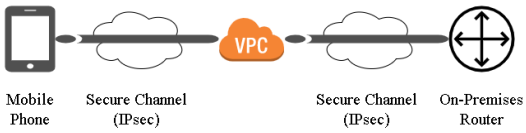

圖 3-2 安全通道網路拓撲

工廠本地端路由器所需的套件包含 strongSwan 及 ip-full，strongSwan 為實作 IPsec XFRM 框架的開源套件，而 ip-full 則包含了 Linux Kernel XFRM 框架的實現，其 IPsec 設置檔如表 3-5。

雲端架構端所使用的作業系統為 Ubuntu Server 20.04，所需的套件僅需 strongSwan，而設置與工廠本地端大抵相同，其 IPsec 設置檔如表 3-6。不過由於 iPhone 在 IPsec VPN 若使用 PSK 認證時，diffie-hellman 僅支援 Group 2 如表 3-7，因此雲端服務端可為行動裝置端所支援的演算法一一配置。

表 3-5 工廠本地端 IPsec 設置檔

conn cloud configuration	
ike	aes-sha1-curve25519
esp	aes-sha1-curve25519
leftid	on-premises
left	192.168.2.1
leftsubnet	192.168.2.0/24
rightid	cloud
right	3.114.X.X
rightsubnet	172.16.0.0/16
auto	route

表 3-6 雲端架構端 IPsec 設置檔

conn mobile configuration	
ike	aes-sha1-modp1024
esp	aes-sha1-modp2048
leftid	cloud
left	172.16.1.52
leftsubnet	172.16.0.0/16
right	%any
rightsourceip	172.15.0.0/16
auto	route
conn on-premises configuration	
ike	aes-sha1-curve25519
esp	aes-sha1-curve25519
leftid	cloud
left	172.16.1.52
leftsubnet	172.16.0.0/16
rightid	on-premises
right	106.104.X.X
rightsubnet	192.168.1.0/24
auto	route

表 3-7 iPhone IPsec 設定與描述[14]

iPhone IPsec 設定與描述	
模式	通道模式
加密演算法	3DES、AES-128、AES256
認證演算法	HMAC-MD5、HMAC-SHA1
Diffie-Hellman 群組	Group 2、Group 5

在 iPhone（iOS 15.3）中設置 IPsec 十分簡單，只需進到 Settings > General > VPN & Device Management > VPN > Add VPN Configuration，便能進行設定，如圖 3-3。

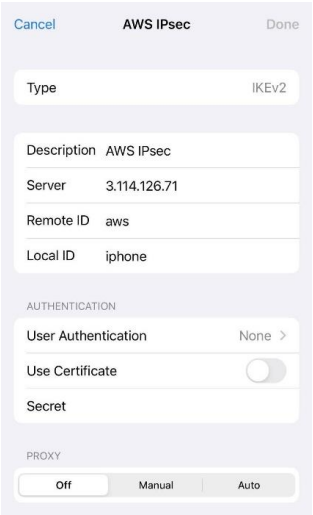

圖 3-3 iPhone 13 IPsec 設定配置

4. 實驗分析

4.1 上雲架構模擬設計

在「工業物聯網架構」中，本研究成功地利用 LinkIt 7688 模擬工業控制器，以上傳資料至雲端服務中的資料存取伺服器，資料格式包含當前時間及狀態。

在「雲端服務架構」中，本研究成功地利用容器化建置資料存取伺服器，接受工廠本地端子網路中的 LinkIt 7688 所上傳的資料，抑或是因應行動裝置端的要求回傳資料。而在行動裝置端開啟 IPsec VPN 時，便能夠經由安全通道連接上資料存取伺服器，伺服器會回傳 LinkIt 7688 所上傳的時間資料，如圖 4-1，而在未開啟 IPsec VPN 時便無法連接上資料存取伺服器。

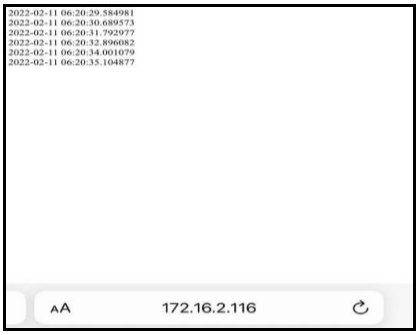

圖 4-1 成功連接資料存取伺服器

4.2 安全通道建置及分析

4.2.1 工廠本地端至雲端架構安全通道

本研究成功地建立與雲端架構端的安全通道，並且透過 iPerf3[15]抽樣 10 天不同時間（早、中、晚），共 30 個樣本的廣域網路、無安全通道站到站（Site-to-Site）、安全通道站到站（Site-to-Site）及點到點（End-to-End）、 LinkIt 7688 乙太網路介面 30 秒平均吞吐率。

而實驗的環境設置如表 4-1 所示，實驗數據採用 iPerf3 Server 端所接受到的 intervals__sum__bits_per_second 為基準。由於 Client 端與 Server 端存在著傳輸延遲，而 iPerf3 所設定的時間間距為 1 秒，故傳送的吞吐率在起初有些許落差，不過在第 1 秒過後便會趨於穩定。

表 4-1 iPerf 環境設置

	iPerf3 Server	iPerf3 Client
廣域網路	112.105.X.X	106.104.X.X
無安全通道站到站	3.114.X.X	106.104.X.X
安全通道站到站	172.16.1.52	192.168.2.1
安全通道端對端	172.16.2.116	192.168.2.100
LinkIt 7688 乙太網路至路由器	192.168.2.1	192.168.2.100

在這個實驗中，從實驗數據圖 4-2 觀察到廣域網路的吞吐率在 20 MBits/s 上下，而無安全通道站到站也在 20 MBits/s 上下，故確保在無安全通道的情況下，從台灣至東京可達到 ISP 所提供的頻寬。

礙於實驗環境之網路頻寬限制，綜括以上實驗數據而言如圖 4-2，在頻寬超過 20 MBits/s 時，首先必須探討廣域網路是否能夠達到 ISP 的理論頻寬，再探討在此頻寬下安全通道的吞吐率是否成為瓶頸，最後在頻寬超過 90 MBits/s 時，LinkIt 7688 的乙太網路吞吐率則為瓶頸，需考慮更換控制器或網路實體介面。

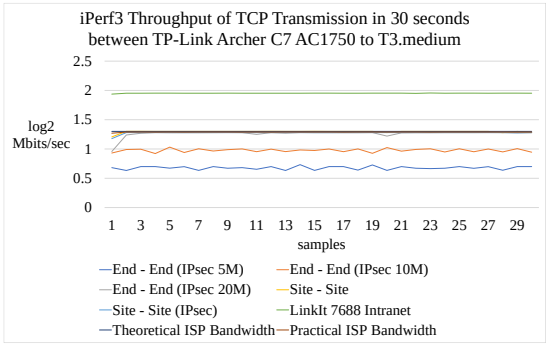

圖 4-2 安全通道吞吐率實驗數據（iPerf3）

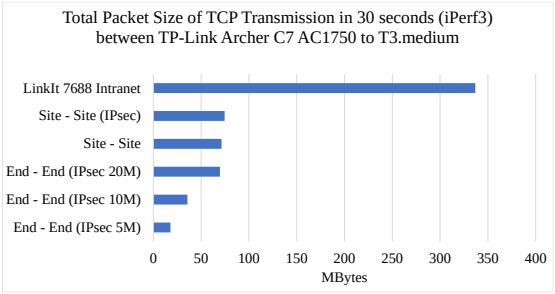

圖 4-3 安全通道封包大小實驗數據（iPerf3）

此外，依實驗數據能夠推論出加解密吞吐率（Encryption/Decryption Throughput）高於網路傳輸吞吐率（Network Throughput），因此安全通道加解密的過程並不會成為網路傳輸之瓶頸，以降低其吞吐率，如圖 4-4。

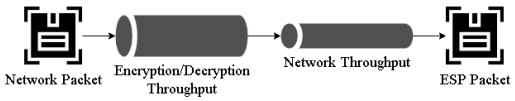

圖 4-4 吞吐率瓶頸示意圖

本研究亦透過 Netperf[16]實測無安全通道站到站（Site-to-Site）、安全通道站到站（Site-to-Site）及點到點（End-to-End）延遲時間。

如表 4-2 所示，這個實驗觀察到無安全通道站到站的平均延遲時間為 40.17 毫秒，安全通道站到站的平均延遲時間為 43.30 毫秒，而點到點的平均延遲時間為 44.14 毫秒，意味著安全通道的處理時間（Encryption/Decryption Processing Delay）約莫 3 毫秒，如圖 4-5。

表 4-2 Netperf 實驗數據

平均延遲時間	最小值	平均值	最大值	標準差
無安全通道	39.32 毫秒	40.17 毫秒	83.20 毫秒	27.50 毫秒
安全通道	40.10 毫秒	43.30 毫秒	87.38 毫秒	32.77 毫秒
點到點	40.97 毫秒	744.14 毫秒	87.61 毫秒	29.24 毫秒

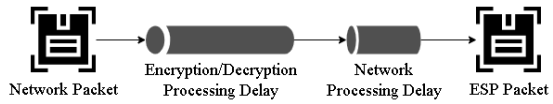

圖 4-5 安全通道處理時間示意圖

4.2.2 iPhone 13 至雲端架構安全通道

本研究亦成功地於 iPhone 13 上使用 iNetTools 進行端對端 ICMP 測試如圖 4-6，並在雲端架構端透過 Wireshark 截取封包如圖 4-7，觀察到封包確實為 ESP，並解密成 ICMP，以達到傳輸安全目的。

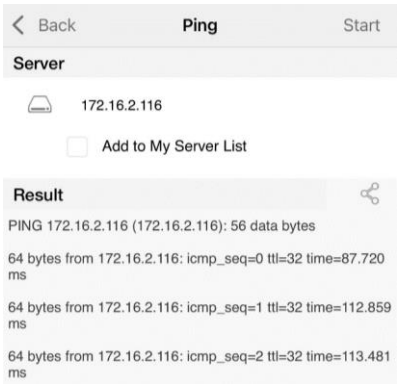

圖 4-6 iPhone 13 安全通道端對端 ICMP 測試

249	28.476529	172.16.1.52	49.217.141.249	ESP
269	29.493975	49.217.141.249	172.16.1.52	ESP
270	29.493975	172.15.0.1	172.16.2.116	ICMP
271	29.494024	172.15.0.1	172.16.2.116	ICMP
272	29.494438	172.16.2.116	172.15.0.1	ICMP

圖 4-7 Wireshark 截取封包

5. 結論

本研究利用 TP-Link Archer C7 1750 及 MediaTek LinkIt 7688 開發版建立簡易工業物聯網環境、Amazon Web Services 作為雲端服務伺服器環境及 Docker、Flask、MongoDB 作為資料存取伺服器，以模擬中小製造業上雲環境。接著使用 iPerf3 和 Netperf 分別分析安全通道吞吐率及延遲，觀察到在 ISP 頻寬 20 Mbits/s 情況下，安全通道不會造成吞吐率瓶頸，處理延遲時間約莫 3 毫秒，同時使用 Wireshark 以驗證安全通道之安全性。

參考文獻

[1] A. Sun, J. Zhou, T. Ji and Q. Yue, "CSB: Cloud service bus based public SaaS platform for small and median enterprises," 2011 International Conference on Cloud and Service Computing, Hong Kong, 2011, pp. 309-314, doi: 10.1109/CSC.2011.6138539.

[2] M. Rao, T. Newe, I. Grout, E. Lewis and A. Mathur, "FPGA Based Reconfigurable IPSec AH Core Suitable for IoT Applications", *13th IEEE International Conference on Pervasive Intelligence and computing (PICom-2015)*.

[3] J. Kirupakar and S. M. Shalinie, "Situation Aware Intrusion Detection System Design for Industrial IoT Gateways," 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), Chennai, India, 2019, pp. 1-6, doi: 10.1109/ICCIDS.2019.8862038.

[4] T. Goethals, D. Kerkhove, B. Volckaert and F. D. Turck, "Scalability evaluation of VPN technologies for secure container networking," 2019 15th International Conference on Network and Service Management (CNSM), Halifax, NS, Canada, 2019, pp. 1-7, doi: 10.23919/CNSM46954.2019.9012673.

[5] S. Jahan, M. S. Rahman and S. Saha, "Application specific tunneling protocol selection for Virtual Private Networks," 2017 International Conference on Networking, Systems and Security (NSysS), Dhaka, 2017, pp. 39-44, doi: 10.1109/NSysS.2017.7885799.

[6] S. Kent and R. Atkinson, "Security Architecture for Internet Protocol", RFC 2401, November 1998.

[7] S. Kent and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[8] S. Kent and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

[9] D. Harkins and D. Carrel, "The Internet Key Exchange", RFC 2409, November 1998.

[10] Rosen, R. (2014). IPsec. In Linux kernel networking implementation and theory (pp. 279–304). essay, Apress.

[11] OpenWRT. OpenWrt Wiki. (2021, September 4). Retrieved February 16, 2022, from https://openwrt.org/

[12] LinkIt™ smart 7688 物聯網開發平台. 聯發科技創意實驗室. Retrieved February 16, 2022, from https://labs..com/zh-tw/platform/linkit-smart-7688

[13] IPsec VPN for Linux, Android, freebsd, Mac OS X, windows. strongSwan. Retrieved February 16, 2022, from https://www.strongswan.org

[14] IPhone 和 iPad 的 Cisco IPsec VPN 設定. Apple Support. Retrieved February 16, 2022, from https://support.apple.com/zh-tw/guide/deployment/depdf31db478/web

[15] GUEANT, V. Iperf - the ultimate speed test tool for TCP, UDP and SCTP. iPerf.fr. Retrieved February 16, 2022, from https://iperf.fr/

[16] Netperf(1) - linux man page. netperf(1): network performance benchmark - Linux man page. (n.d.). Retrieved February 16, 2022, from https://linux.die.net/man/1/netperf