科技部補助專題研究計畫成果報告

（□期中進度報告/☑期末報告）

一個利用系統時間解析度的時脈偏移複製攻擊偵測方法

計畫類別：☑個別型計畫　□整合型計畫
計畫編號：MOST　104　－　2221　　－　E　－　011　　－　070　　－
執行期間：　104　年　8　月　1　日至　105　年　7　月　31　日

執行機構及系所：國立台灣科技大學資訊工程系

計畫主持人：鄧惟中
共同主持人：
計畫參與人員：Komang Oka Saputra、江奇峰、蘇育毅

本計畫除繳交成果報告外，另含下列出國報告，共 ___ 份：
□執行國際合作與移地研究心得報告
□出席國際學術會議心得報告

期末報告處理方式：

1. 公開方式：
   □非列管計畫亦不具下列情形，立即公開查詢
   □涉及專利或其他智慧財產權，□一年□二年後可公開查詢
2. 「本研究」是否已有嚴重損及公共利益之發現：□否 □是
3. 「本報告」是否建議提供政府單位施政參考 □否 □是，___
   （請列舉提供之單位；本部不經審議，依勾選逕予轉送）

中　華　民　國　105　年　6　月　13　日

中文摘要

針對使用時脈偏移裝置識別技術的識別系統，攻擊者可能會發送偽造的時間戳記進行時脈偏移複製攻擊。在本研究的實驗中，確認了偽造的時脈偏移與目標偏移之間的誤差常在±1 ppm 內，若識別系統只用時脈偏移值的結果來認證使用者，攻擊者將有機會通過認證。然而，不同的作業系統具有不同的時間解析度，攻擊者所使用的作業系統會影響其實作複製攻擊的方式。我們發現當攻擊者端的時間解析度僅為毫秒等級時，可以在偏移數據中發現以固定頻率出現的跳躍點，其規模近似於攻擊者的系統時間解析度。我們利用這個明顯的攻擊特徵，偵測每個跳躍點在數據集中出現的位置，並移除這些跳躍點的影響來重組攻擊者原始的時脈偏移值。如果偵測前後時脈偏移值的誤差超過 1 ppm，我們就會判定對方為攻擊者。經由實驗結果分析，攻擊者使用設備的時脈偏移值與重組後估計的時脈偏移值其誤差皆在 1 ppm 以內，而偽造的時脈偏移與重組後的時脈偏移誤差皆在 1 ppm 以上，因此可以有效偵測並阻絕時脈偏移的複製攻擊。

關鍵字:時脈偏移、複製攻擊、系統時鐘

英文摘要

The clock skew, or the physical ticking rate difference between two digital clocks, has been revealed to have potential on serving as the device fingerprint for identification/authentication purpose. However, it remains as an open issue to detect clock skew replication behavior, which is realized by sending altered timestamps. In this study, it is confirmed that an attacker may fake any target skew with the error being no more than 1 ppm in local network environments. Besides, it is also observed that the value of fake timestamps are affected by the time resolution of the attacker's system clock. When the resolution is 1 ms or lower, a relatively large jump between consecutive offsets happens regularly, and the scale of each jump is theoretically the very time resolution of the attacker's system clock. This characteristic is thus adopted to develop a filtering method such that the receiver is able to detect fake timestamps. When the periodical jumps are detected, the filter module abandons these jumps to recover the original clock skew. Experimental results on 15.6 ms and 1 ms time resolutions show that the developed method is effective to detect skew replication attacks, and the errors of the recovered clock skews are no more than 1 ppm from the real skews of the attackers.

Index Terms—clock skew; replication attack; time resolution

報告內容

前言

在雲端服務已經相當成熟且蓬勃發展的環境下，使用者能運用的雲端資源愈來愈豐富，加上智慧型手機與平板電腦等行動裝置迅速普及，只要使用者具備行動裝置即可隨時隨地存取這些雲端服務，使用者本身與其所使用的裝置已經有著密不可分的關係。雖然這些進步使得資訊應用愈來愈方便，但敏感性資訊也比以前相對透明許多，為了持續加強資訊安全的強度，本研究利用實體裝置獨特的物理特性:時脈偏移(Clock Skew)，來識別使用者與其所使用的裝置身份，進一步防止帳號竊取與裝置盜用的狀況發生。

對於一台實體裝置，有許多屬性可以協助我們做為識別該裝置的依據，例如 IP 位址、MAC 位址或是 Cookie 等等，但這些屬性都具有弱點，就是容易被偽造、缺乏獨特性以及在不同環境下都有所不同，使得這些屬性不足以作為一台實體裝置真正的指紋。然而，雖然時脈偏移是與硬體有關的屬性，卻與 MAC 位址的意義大相逕庭。MAC 位址是在產品製造時，由製造者特別賦予給產品的序號或身分證號碼，但這個號碼是很容易被偽造的；而時脈偏移又稱做時鐘偏斜，意指任意兩台實體裝置中所記錄的時間，隨著真實時間過去產生愈差愈大的現象，這是由於電腦主機板上負責計時的 BIOS 時鐘，其石英震盪器(Crystal Oscillator)震盪頻率的偏差所導致，此偏差主要來自於震盪器在製作過程中產生的誤差，使得轉換後的系統時間與標準時間相比經常有快慢不一的現象。除非深知裝置之間在時鐘震盪頻率上的偏差，否則偽造時脈偏移相當具有難度。

一般來說，時脈偏移具有兩個特性，第一，兩個裝置之間的時脈偏移在正常溫度下是相對穩定的；第二，任意兩個裝置之間的時脈偏移在精確度達到 1 ppm(Parts Per Million)時是可以清楚分辨的。基於這兩個特性，時脈偏移可以被視為任何擁有數位時鐘裝置的指紋。

偽造時脈偏移的方法

在過去的研究中，Huang[3]、Moon[4]、Kohno[1]等人皆使用線性規劃法(Linear Programming Method, LPM)來量測時脈偏移。LPM 是根據線性的偏移量集合所形成的上或下界來量測時脈偏移。本研究使用[4]的概念，利用偏移量集合的下界來估算時脈偏移。

在一組由量測者的時間戳記$t_m(i)$以及與受測者之間的偏移量o_i'所組成的大小為 n 的序列$(t_m(1),o_1')$，...，$(t_m(n),o_n')$中，假設有一條線

y=αx+β，α 為該線之斜率，β 為該線在 Y 軸上的截距。則時脈偏移的估算必須滿足：

$$\forall i = 1 \ldots n, \alpha \cdot t_m(i) + \beta \leqq o_i' \text{(式 1)}$$

式 1 代表解答必須是所有$(t_m(i), o_i')$的下界。並且需使用線性規劃來最小化下面的表示式：

$$\frac{1}{n}\sum_{i=1}^{n}(o_i' - (\alpha \cdot t_m(i) + \beta)) \text{(式 2)}$$

此問題可以使用[4]所提到的雙變量下之線性規劃法，從而計算出時脈偏移值 α。

鄭理介[12]的研究指出了 LPM 相較其他量測演算法的優點，由於 LPM 產生的結果線段會落在所有偏移量之下，即使數據參雜大量離群值(outlier)，LPM 也能正確地估算時脈偏移。如圖 1 所示，紅色線段為使用 LPM 量測的結果。

本研究後續的實驗皆是採取一次性量測的非即時量測，即使我們在穩定的區域網路中進行實驗，仍可能出現少數規模較小的離群值，為了讓結果盡可能不被離群值影響，基於 LPM 的穩定性，最終我們選擇 LPM 作為本研究量測時脈偏移的方法。

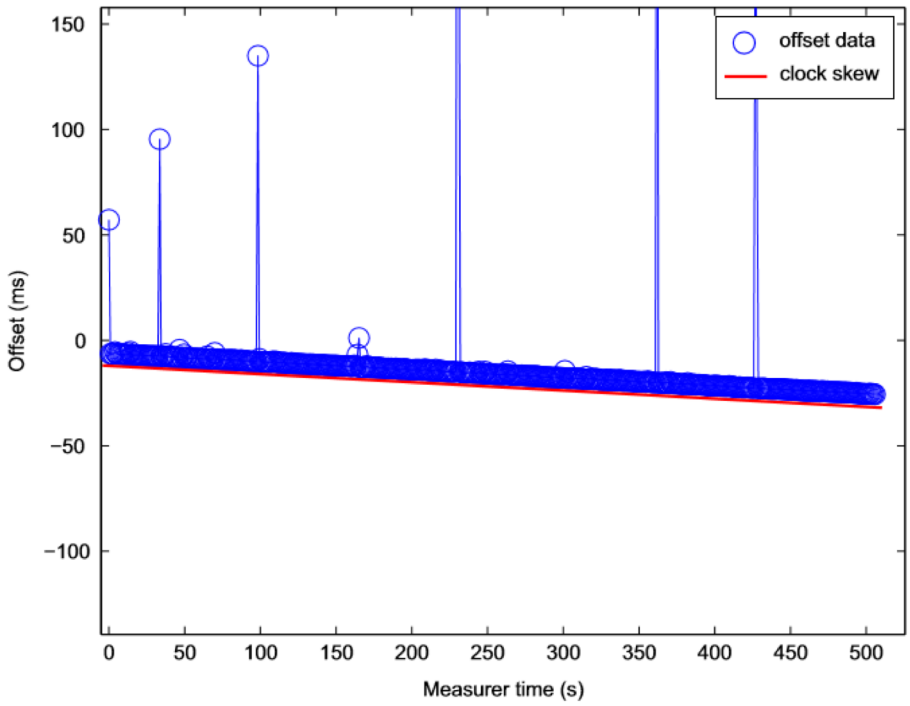

圖 1：LPM 的量測結果

偽造時脈偏移

在 Huang 等人[2]的研究中曾提及，任意兩個節點 A 與 B，或任意三個節點 A、B 與 C 之間的時脈偏移會具有以下關係：

1. 加法反元素(Additive inverse)

$$\text{skew}_{AB} \cong -\text{skew}_{BA} \qquad (式\ 3)$$

2. 線性關係(Linearity)

$$\text{skew}_{AB} + \text{skew}_{BC} \cong \text{skew}_{AC} (式\ 4)$$

因為時脈偏移值在量測上會有誤差，兩式中的結果只能達到近似，無法完全相等。利用這兩項關係，Huang 等人[2]讓攻擊節點進行逆向工程，成功偽造另一個正常節點的時脈偏移。

時間戳記與發送間隔分析

假設攻擊者 A 知道 A 對於受害者 T 的時脈偏移：s_{TA}，且 A 試圖欺騙量測者 M 誤以為 A 是 T。每當 A 經過一秒視同 M 經過了 $1+s_{AM}$ 秒，且s_{AM}與s_{TM}有以下關聯性：

$$r = \frac{1 + s_{TM}}{1 + s_{AM}} \qquad (式\ 5)$$

其中r為一恆定值。

令t_1代表第一筆時間戳記，亦或是第一個封包的發送時間，所以第 i 個封包的時間戳記應為：

$$t_i' = t_1 + r(t_i - t_1) \qquad (式\ 6)$$

假設 A 不知道s_{AM}，則 A 可透過以下等式得知取得s_{AM}的近似值：

$$1 + s_{TM} = (1 + s_{TA})(1 + s_{AM}) \simeq 1 + s_{TA} + s_{AM} \quad (式\ 7)$$

或可整理成$s_{TM} \simeq s_{TA} + s_{AM}$之形式。因時脈偏移之絕對值不會超過 200 ppm[16]，因此兩時脈偏移值之乘積可忽略之。

此攻擊手法看似可行，但有一明顯弱點：M 可要求以不同週期接收時間戳記。因任二時間t_i及t_{i+1}間應有固定間隔Δt，所以 A 無法在尚未得知新發送週期的情況下偽造t_{i+1}。因此替代偽造時脈偏移之方案為將發送週期改為$r^{-1}\Delta t$。

舉例來說，假設攻擊者 A 已知s_{TM}為 20 ppm，s_{AM}為 200 ppm，且量測者 M 要求發送週期為 1 秒。以上情境可看出 A 的時脈明顯快於受

害者 T，T 的時脈僅略快於 M。A 為了達到每秒發送間隔為 1 秒之要求必須改變自身的發送週期，以 A 之觀點來看發送間隔應為 $1 \cdot (1+200 \cdot 10^{-6})/(1+20 \cdot 10^{-6}) \approx 1.00018$ 秒。

系統時間解析度之限制

時間解析度代表一個系統更新其時鐘的最小單位，這將決定我們能夠以多小的時間間隔進行事件的觸發。因此上述方法之發送間隔應在系統時間解析度的範圍內。時間解析度會因作業系統而異，例如 Linux 及 Android 為 1 μs，微軟 Windows 系統則為 15.6 ms。然而，Windows 系統的使用者能以 system call 之形式將系統解析度提高至 1 ms 以符合研究需求。

令 k 代表攻擊者 A 的時間解析度，且一程序之接收間隔為Δt秒，亦即程序會每隔$\left\lceil \frac{\Delta t}{k} \right\rceil \cdot k$秒接收一次時間戳記。為求方便，我們在此假設k能整除Δt。當 A 將自身的發送間隔設定為$r^{-1}\Delta t$時，實際的發送間隔會變成$\left\lceil \frac{r^{-1}\Delta t}{k} \right\rceil \cdot k$秒，偽造出來的時間戳記則會每次多出$\Delta t$秒之差。誤差$e = r^{-1}\Delta t \bmod k$或許不大，但長久累積下來卻能影響時脈偏移的結果。為了修正誤差，A 必須要對每$\left\lceil \frac{k}{e} \right\rceil$次發送之時間戳記作補償，方法如下所示：

$$t_{i+1} = \begin{cases} t_i + \left\lceil \frac{r^{-1}\Delta t}{k} - 1 \right\rceil \cdot k \,, if \left\lceil \frac{k}{e} \right\rceil \mid i \\ t_i + \left\lceil \frac{r^{-1}\Delta t}{k} \right\rceil \cdot k \,, otherwise \end{cases} \quad (式\ 8)$$

$$t'_{i+1} = t'_i + \Delta t \quad (式\ 9)$$

偵測偽造的時脈偏移

根據(式 8)及(式 9)，A 之時間解析度為k，M 收集之時間戳記集合有一明顯特徵：數據分布圖之 Y 軸代表“M 之時間值減去 A 之時間值”，故每$\left\lceil \frac{k}{e} \right\rceil$次皆會因(1)之修正動作而產生一幅度為$k$之跳躍點(jump

point)。在分布圖中跳躍點易與因傳輸延遲造成的離群值混淆，但跳躍點有一定規律性不似隨機出現的離群值，因此 M 可以藉由比較方式判斷跳躍點的出現與否。

原本 A 採用跳躍點攻擊手法偽造 T，M 可藉由分析跳躍點並切割成數個子圖，最後重組成原始分布圖得到 A 的原始時脈偏移值方能破解攻擊手段，維持系統的安全性。Algorithm 1 為一複製攻擊的過濾方法，O 為 A 與 M 之時間戳記差值集合；m則為過濾閥值，等同攻擊者的時間解析度；LargerDelay用來儲存大於m之數據；JumpPoint儲存攻擊時的起始點，亦等同跳躍點位置；idx則為LargerDelay和JumpPoint的索引。演算法第三行至第六行將疑似攻擊點的位置紀錄到LargerDelay中，由於離群值與跳躍點都會被考慮到，因此第八行至第十三行將LargerDelay內的點作篩選並將正確的跳躍點紀錄到JumpPoint中。最後第十四至第十九行將跳躍點移除、修正時間戳記差值，達到破解攻擊者試圖改變時脈偏移值之目的。還原後的分布圖即可量測攻擊者 A 對量測者 M 之時脈偏移，不僅能成功阻擋攻擊更能辨識攻擊者 A 之身分。

Algorithm 1 Clock skew replication attack filter

Require: O, m

1: $LargerDelay$ = $null$
2: $JumpPoint$ = $null$
3: **for** $i = 1; i \leq O.length;\ i++$ **do**
4: **if** $The\ Absolute\ value\ of\ (O_{i+1} - O_i) \geq m$ **then**
5: Recording $(i+1)$ to $LargerDelay$
6: **end if**
7: **end for**
8: **for all** $idx \in LargerDelay$ **do**
9: Finding each continuous idx and then grouping them as $[first\ idx,\ last\ idx]$
10: **if** $first\ idx\ ==\ last\ idx$ **then**
11: Recording idx to $JumpPoint$
12: **end if**
13: **end for**
14: **for all** $idx \in JumpPoint$ **do**
15: O_{temp} = $O_{idx} - O_{idx-1}$
16: **for** $j = idx;\ j \leq JumpPoint.length;\ j++$ **do**
17: O_j = $O_j - O_{temp}$
18: **end for**
19: **end for**

研究成果

實驗環境

本研究的實驗環境將使用一台固定的個人電腦當作接收端，也就是量測者 M；一台筆記型電腦當作發送端，也就是攻擊者 A。發送端部分，我們在筆記型電腦上架設了雙系統，分別為微軟的 Windows7 作業系統(32 位元)，以及 Ubuntu 作業系統(32 位元)；而接收端則是使用 Ubuntu 14.04 作業系統(64 位元)。

硬體規格方面，發送端的筆記型電腦為 Asus Notebook M50SV，其 CPU 規格為 Intel(R) Core(TM) Duo CPU T9300 2.5GHz，RAM 為 4GB；接收端的個人電腦 CPU 規格為 Intel(R) Core(TM) Duo CPU E8300 2.83GHz，RAM 為 4GB。

軟體部分，發送端與接收端的程式皆是以 Python 程式語言撰寫，版本為 3.4.3，使用的時間函數為標準函式庫的 time.time()函數，用來獲取系統時間。

時脈偏移複製攻擊實驗

首先我們針對採用 Windows 作業系統之攻擊者作時脈偏移複製攻擊，其時間解析度為 15.6 ms。如圖 1 所示，攻擊者 A 原始時脈偏移 s_{AM}為-15.5 ppm。

現在攻擊者要偽造-215.5 ppm、-35.5 ppm，以及-18.5 ppm 之時脈偏移，為了達成目的，攻擊者必須加快自身之時脈偏移約 200ppm，20 ppm，以及 3 ppm 以符合要求。圖二為實作複製攻擊偽造時脈偏移為-215.5 ppm 之數據分布圖，經 LPM 量測結果時脈偏移為-215.79 ppm。

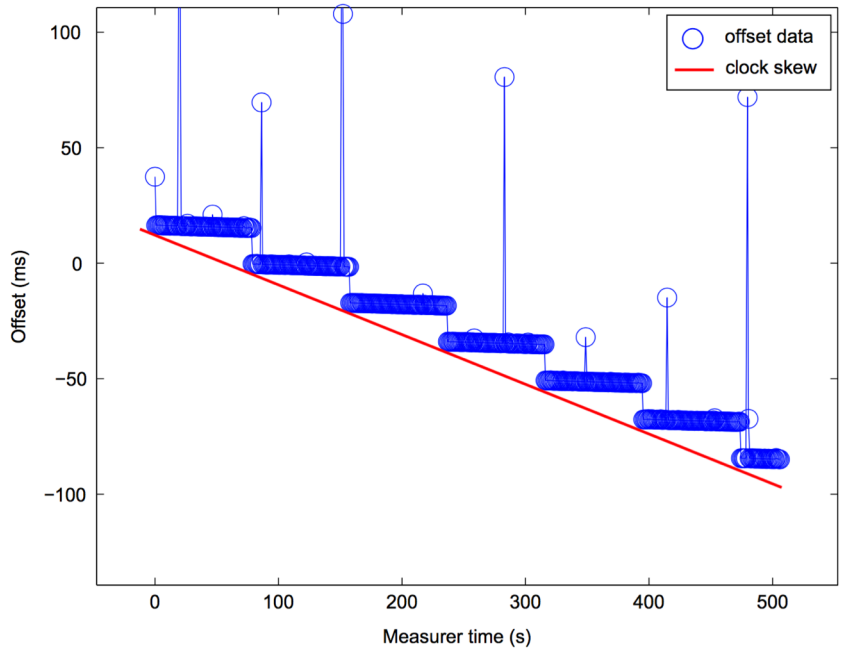

圖 2：解析度為 15.6 ms 之攻擊者偽造時脈偏移為-215.5 ppm 之結果，經 LPM 量測結果為-215.79 ppm

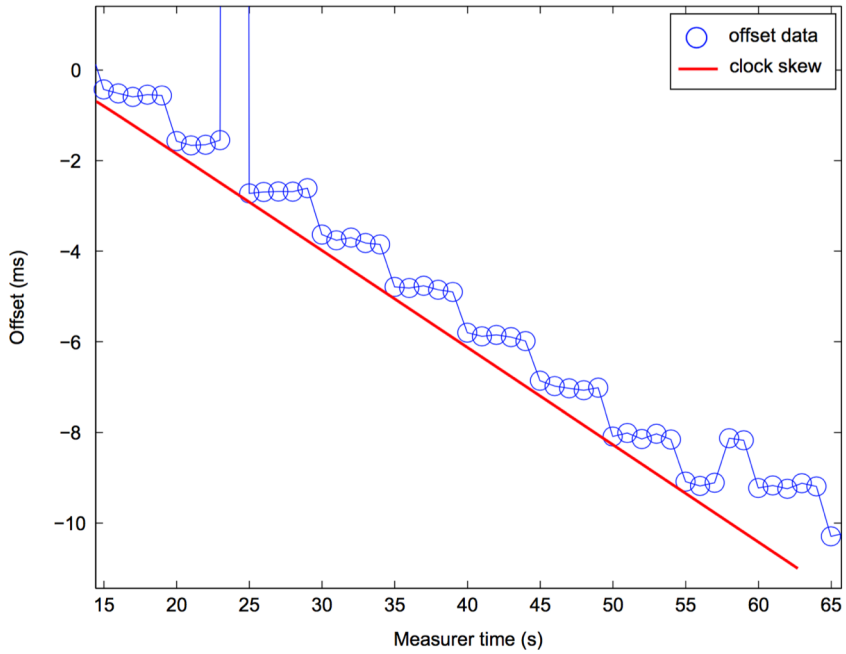

圖 3：解析度為 1 ms 之攻擊者偽造時脈偏移為-215.5 ppm 之結果，經 LPM 量測結果為-214.83 ppm

從圖 2 我們可以發現跳躍點明顯地將分布圖區分成數個部分，且每次跳躍的距離約等同 15.6 ms，恰為攻擊者之時間解析度；圖 3 分布特性與圖 2 相當，但跳躍距離則為 1 ms。本複製攻擊實驗誤差僅在 0.67 ppm 以內，表示攻擊者能藉由此方法偽造精準的時脈偏移值。

表1列出了各類情況之實驗結果，實驗命名方式為:W代表Windows作業系統，L代表Ubuntu 14.04，15m代表攻擊者具15.6 ms之時間解析度，1m代表1 ms，1μ代表1 μs；200p代表攻擊者欲增加200 ppm之時脈偏移，20p代表20 ppm，3p代表增加3 ppm之幅。

實驗組合	初始時脈偏移 (ppm)	目標時脈偏移 (ppm)	實作結果 (ppm)	誤差 (ppm)
W15m200p	-15.5	-215.5	-215.79	0.29
W1m200p			-214.83	0.67
L1μ200p			-214.96	0.54
W15m20p		-35.5	-35.33	0.17
W1m20p			-35.37	0.13
L1μ20p			-35.48	0.12
W15m3p		-18.5	-18.69	0.19
W1m3p			-18.12	0.38
L1μ3p			-18.58	0.08

表 1：時脈偏移複製攻擊實驗結果

實驗組合	偵測時脈偏移 (ppm)	還原時脈偏移 (ppm)	與初始值之誤差 (ppm)
W15m200p	-215.79	-15.53	0.03
W15m20p	-35.33	-15.21	0.29
W15m3p	-18.69	-15.57	0.3
W1m200p	-214.83	-16.07	0.57
W1m20p	-35.37	-15.91	0.41
W1m3p	-18.12	-14.97	0.53

表 2：時脈偏移複製攻擊過濾實驗結果

從表1我們能清楚得知無論攻擊者使用Windows作業系統(時間解析度為1 ms)或Linux作業系統(時間解析度為1 μs)皆能在1 ppm之誤差內成功偽造各種時脈偏移。

此外在不同時間解析度之環境下跳躍點出現週期亦不相同。從圖2和圖3可以觀察到在時間解析度為15.6 ms之環境下每隔78秒會出現一次跳躍點；在時間解析度為1 ms之環境下則是每隔5秒即出現一次跳躍點，量測時間愈久，跳躍點出現次數愈多。因此，我們可就不同的量

測時間觀察跳躍點對時脈偏移的影響，最短30秒，最長可達1000多秒，量測時間決定了複製攻擊是否成功。

複製攻擊過濾實驗

由於跳躍點只會在時間解析度為毫秒等級之 Windows 作業系統下發生，故本研究之複製攻擊過濾實驗僅針對時間解析度為 15.6 ms 及 1 ms 之情形實作與探討。同表 1 之命名方式，表 2 亦列出各類情況之實驗及其結果，我們發現本過濾法能成功還原攻擊者之原始時脈偏移以達到防禦之目的。與原始時脈偏移值-15.5 ppm 相較，在 W1m200p 之情況下有最大誤差 0.57 ppm。

圖 4 在 W15m200p 之情境顯示本過濾法去除跳躍點後產生一新的分布圖(如圖粉色部分所示)，其時脈偏移約與攻擊者之原始時脈偏移相同，誤差僅在 1 ppm 之內，表示複製攻擊能被有效破解且能反偵測攻擊者之時脈偏移資訊。

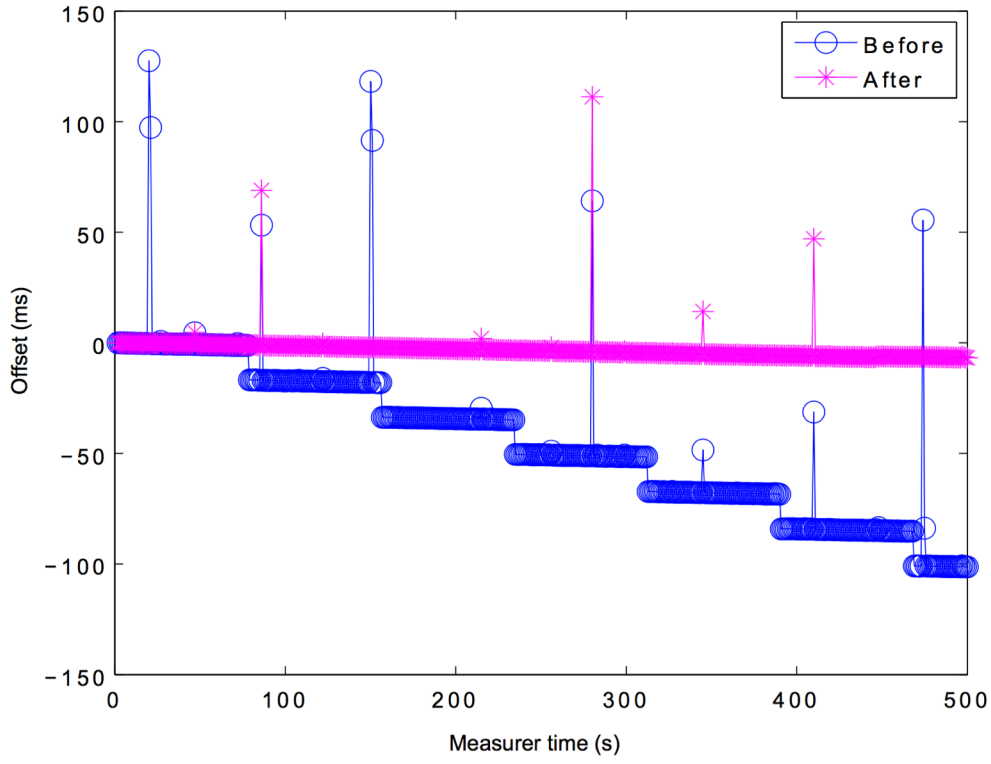

圖 4：複製攻擊過濾實驗結果，藍色部分時脈偏移為-215.79 ppm，經還原後粉色部分時脈偏移為-15.53 ppm

結果與討論

在本研究中，惡意的攻擊者可以藉由偽造發送週期以及時間戳記的方式修改自身的時脈偏移來偽造其他使用者的身分，在此方法下，偽造的時脈偏移結果與我們預期的結果差距皆在±1 ppm，若識別系統只用時脈偏移值的結果來認證使用者，攻擊者將會通過認證。然而，不同的作業系統具有不同的時間解析度，攻擊者所選用的作業系統將會影響其實作複製攻擊的方式。我們發現當攻擊者端的時間解析度僅為毫秒級時，可以在偏移數據中發現以固定頻率出現的跳躍點，且規模約等於攻擊者的解析度。對於這個明顯的攻擊特徵，我們在實驗過程中偵測跳躍點在數據中出現的位置，並移除跳躍點對後續數據的影響來重組攻擊者原始的偏移值，最後透過前後時脈偏移值的誤差來判定客戶端是否為攻擊者。本研究的目的在於驗證一些攻擊手法在基於時脈偏移裝置識別技術上的可行性，使識別系統的防禦措施能夠愈趨完整，若能搭配其他認證方式作為後盾，攻擊者將無法輕易通過認證。

參考文獻

[1] Tadayoshi Kohno, Andre Broido,and Kc Claffy, “Remote Physical Device Fingerprinting,” IEEE Transactions on Dependable and Secure Computing, vol. 2, no. 2, pp. 93–108, 2005.
[2] Ding-Jie Huang and Wei-Chung Teng, “A Defense Against Clock Skew Replication Attacks in Wireless Sensor Networks,” Journal of Network and Computer Applications, Elsevier, vol. 39, pp. 26-37, 2013.
[3] Ding-Jie Huang, Kai-Ting Yang, Chien-Chun Ni, Wei-Chung Teng, Tien-Ruey Hsiang, and Yuh-Jye Lee, “Clock Skew Based Client Device Identification in Cloud Environments,” The 26th IEEE International Conference on Advanced Information Networking and Applications (IEEE AINA 2012) , pp. 526-533, 2012.
[4] Sue B. Moon, Paul Skell, and Don Towsley, “Estimation and Removal of Clock Skew from Network Delay Measurements,” IEEE INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings vol. 1, pp. 1-25, 1999.
[5] Vern Paxson, “On Calibrating Measurements of Packet Transit Times,” ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems (SIGMETRICS 1998/PERFORMANCE

1998), vol. 26, no. 1, pp. 11-21, 1998.
[6] Steven J. Murdoch, “Hot or not: Revealing Hidden Services by Their Clock Skew,” in Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 27-36, 2006.
[7] Sebastian Zander and Steven J. Murdoch, “An Improved Clock-skew Measurement Technique for Revealing Hidden Services,” in Proceedings of the 17th conference on Security symposium, pp. 211-225, 2008.
[8] Ding-Jie Huang, Wei-Chung Teng, Chih-Yuan Wang, Hsuan-Yu Huang, and Joseph M. Hellerstein, “Clock Skew Based Node Identification in Wireless Sensor Networks,” IEEE Global Communications Conference(GLOBECOM 2008), pp. 1-5, 2008.
[9] Suman Jana and Sneha Kumar Kasera, “On Fast and Accurate Detection of Unauthorized Wireless Access Points Using Clock Skews,” in Proceedings of the 14th ACM International Conference on Mobile Computing and Networking, pp. 104-115, 2008.
[10] Swati Sharma, Alefiya Hussain, and Huzur Saran, “Experience with Heterogenous Clock-skew Based Device Fingerprinting,” in Proceedings of the 2012 Workshop on Learning from Authoritative Security Experiment Results (LASER 2012), pp. 9-18, 2012.
[11] Chrisil Arackaparambil, Sergey Bratus, Anna Shubina, and David Kotz “On the Reliability of Wireless Fingerprinting Using Clock Skews.” in Proceedings of the third ACM conference on Wireless network security (WiSec 2010), pp. 169-174, 2010.
[12] 鄭理介(2013)。《基於時脈偏移的可攜式裝置識別技術》。國立臺灣科技大學碩士論文。
[13] System Time Wikipedia：https://en.wikipedia.org/wiki/System_time, Feb 2015.
[14] Microsoft Official Website：https://www.microsoft.com/en-us, Feb 2015.
[15] QPython Official Website：http://qpython.com/, June 2015.
[16] K. Oka Saputra, W.-C. Teng, and T.-H. Chen, “Hough transform-based clock skew measurement over network,” IEEE Trans. Instrum. Meas., 2015, published online, to appear in print.