行政院國家科學委員會專題研究計畫 成果報告

異質多網安全檢測平台建置計畫(III)
研究成果報告(完整版)

計 畫 類 別 ：個別型
計 畫 編 號 ：NSC 100-2219-E-009-005-
執 行 期 間 ：100年01月01日至101年04月30日
執 行 單 位 ：國立交通大學資訊工程學系（所）

計 畫 主 持 人：謝續平
共 同 主 持 人：曾文貴
計畫參與人員：學士級-專任助理人員：郭明華
學士級-專任助理人員：林慧雯
碩士班研究生-兼任助理人員：許基傑
碩士班研究生-兼任助理人員：梁偉明
碩士班研究生-兼任助理人員：黃韋翔
碩士班研究生-兼任助理人員：許鴻生
碩士班研究生-兼任助理人員：葉書宏
碩士班研究生-兼任助理人員：籃日全
碩士班研究生-兼任助理人員：彭日伸
碩士班研究生-兼任助理人員：盧豔銘
碩士班研究生-兼任助理人員：陳彥宇
碩士班研究生-兼任助理人員：廖政博
碩士班研究生-兼任助理人員：張家愷
碩士班研究生-兼任助理人員：賴託登
碩士班研究生-兼任助理人員：梁喬峰
碩士班研究生-兼任助理人員：黃冠霖
碩士班研究生-兼任助理人員：朱慶峰
碩士班研究生-兼任助理人員：蘇修醇
碩士班研究生-兼任助理人員：石穎
碩士班研究生-兼任助理人員：鍾凱任
碩士班研究生-兼任助理人員：鄭昀旻
博士班研究生-兼任助理人員：陳柏廷
博士班研究生-兼任助理人員：沈宣佐

博士班研究生-兼任助理人員：何秉哲
博士班研究生-兼任助理人員：王繼偉

報告附件：國外研究心得報告
出席國際會議研究心得報告及發表論文

公開資訊：本計畫可公開查詢

中華民國 101年05月30日

中文摘要： 依據行政院『我國資通安全政策白皮書』、國外經驗與借鏡，設置國家級之「資通安全研究與教學中心(TaiWan Information Security Center, TWISC)」，在中央研究院李德財院士的領導下，於94年成立台灣科技大學心資通安全研究與教學中心(TWISC@NTUST)、交通大學資通安全研究與教學中心(TWISC@NCTU)、與成功大學資通安全研究與教學中心(TWISC@NCKU)，分別從事其專長領域研究、技術建置與推廣。

TWISC@NCTU在本計畫主要的目的在於建置一個異質多網安全檢測平台、開發與建置產官學研所需的安全檢測工具，以及提供政府機關、產業界或是財團法人異質多網安全檢測的服務。希望藉由我們所開發各種安全檢測工具來發現異質多網與行動設備潛在的安全問題，讓管理人員以及一般使用者可提早修補漏洞、改善問題以提高行動設備安全性。為了在2011年開發適合產官學研的安全檢測工具，我們已經與多個政府機關、法人以及產業界建立產學合作關係，包括：總統府國家安全會議、法務部調查局、工研院、資策會、行政院研考會、國家資通安全會報技術服務中心、中華電信、友訊科技、宏達電、趨勢科技、喬鼎科技、中科院、教育部等，並了解他們的檢測需求。舉例說明：我們和工研院資通所達成合作共識，開發Android相關安全檢測工具。此外，我們也和宏達電、友訊科技及中華電信合作，執行軟體安全檢測以及惡意程式行為分析。這些單位在Access Point/router、智慧型手機也有滲透檢測之需求。在2011年，我們針對上述單位的需求來開發適當且客製化的檢測工具。此外，我們也積極地與其他單位聯繫以及交流，來開發出更符合產研單位需求的安全檢測工具。

在2011年，本計畫開發了7個全新的安全檢測防護工具(請見表1)，且繼續客制化與維護已開發完成的15個檢測工具。同時，我們也持續把適切的檢測工具轉成線上服務，讓更多人可因此受惠。藉由此平台的建置與檢測工具的開發，我們希望提供政府機關、財團法人及高科技廠商網路安全檢測的服務，並且技轉所開發的檢測工具，以幫助上述單位發現漏洞及弱點。如此一來將可提高產業的經濟效益、提升無線產品附加價值、節省因網路攻擊或系統弱點所消耗的產值、節省專業檢測人力並且有效減少各種有線、無線網路環境的攻擊。

此外，在2011年我們的成果包含技術移轉1件、先期技術移轉1件、技術服務1件、與產學合作達12件，總金額超越本計畫書所規畫之400萬元。此項成果說明本計畫之建置成果在產業界之可應用性與前瞻性。本計畫之學術以及產業研究

均有相當成果。參與本計畫之成員於2011年發表於國際重要期刊之論文數共19篇，發表於國際研討會之論文數共16篇以及國內研討會之論文共2篇。2011年美國專利獲證5件，提出美國以及台灣專利申請各1件。詳細列表請參閱（七）計畫成果。

中文關鍵詞： 網路安全、系統安全、軟體安全、惡意程式、社群網路

英文摘要： The goal of TWISC@NCTU is to develope a heterogeneous network security inspection platform and security tools applied in academic, industry and government in practice. This project can provide the services of heterogeneous network security analysis for any cooperative organization. In 2011, we cooperated with National Security Council, Ministry of Justice, ITRI, III, REDC, ICST, Chunghwa Telecom, D-Link, HTC, Trend Micro, Promise, CSIST, Ministry of Education, etc. The project meets the requirements, which were suggested by the cooperative organizations, to disclosure the vulnerabilities of mobile equipment for security improvement. For example, is one of the work items due to the shared view reached with Information and Communications Research Laboratories, ITRI. Additionally, both testing software security and analyzing behaviors of malware in our research comply with the industry demand by HTC, D-Link and Chunghwa Telecom.

In 2011, 7 new security testing tools are proposed and implemented with the functionalities required by industry and government, and some functionalities of our project are appropriately turned into on-line services for benefiting the people who are interested in it. With the platform and tools, we anticipate enhancing information security in government sectors, corporations, and hi-tech industries, and we conucted technology transfer with related companies to discover security vulnerabilities in advance. In this way, the quality of network products is increased； the manpower dealing with security threats is reduced； the system vulnerabilities are discovered； finally, the threats in heterogeneous networks are eliminated. Moreover, our team conducted 16 remarkable cooperation projects, which includes industry-university cooperation, technical assistance services, and software licensing, that meet our project goal of 4 million dollars in revenue. This result shows the applicability of our novel techniques. In publication, we have published 7 journal papers, 11 international conference papers, and 1 national conference papers. We also received 3 patents and filed two new patents. Thus, this demonstrated our accomplishments in both academics and industry.

英文關鍵詞： Network Security, System Security, Software Security, Malware, Social Network.

行政院國家科學委員會補助專題研究計畫 ■ 成果報告 □期中進度報告

異質多網安全檢測平台建置計畫(III)

計畫類別：■ 個別型計畫　　□ 整合型計畫
計畫編號：NSC100-2219-E-009-005-
執行期間：100 年 1 月 1 日至 100 年 12 月 31 日

計畫主持人：謝續平
共同主持人：曾文貴
協同主持人: 黃世昆、黃育綸、趙禧綠、吳育松，孫宏民

成果報告類型(依經費核定清單規定繳交)：□精簡報告　■完整報告

本成果報告包括以下應繳交之附件：
■赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
■出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢
□涉及專利或其他智慧財產權，□一年□二年後可公開查詢

執行單位：國立交通大學資訊工程學系
資通安全研究與教學中心 TWISC@NCTU

中華民國　101　年　04　月　25　日

摘要

依據行政院『我國資通安全政策白皮書』、國外經驗與借鏡，設置國家級之「資通安全研究與教學中心(TaiWan Information Security Center, TWISC)」，在中央研究院李德財院士的領導下，於94年成立台灣科技大學心資通安全研究與教學中心(TWISC@NTUST)、交通大學資通安全研究與教學中心(TWISC@NCTU)、與成功大學資通安全研究與教學中心(TWISC@NCKU)，分別從事其專長領域研究、技術建置與推廣。

TWISC@NCTU 在本計畫主要的目的在於建置一個異質多網安全檢測平台、開發與建置產官學研所需的安全檢測工具，以及提供政府機關、產業界或是財團法人異質多網安全檢測的服務。希望藉由我們所開發各種安全檢測工具來發現異質多網與行動設備潛在的安全問題,讓管理人員以及一般使用者可提早修補漏洞、改善問題以提高行動設備安全性。為了在 2011 年開發適合產官學研的安全檢測工具，我們已經與多個政府機關、法人以及產業界建立產學合作關係，包括：總統府國家安全會議、法務部調查局、工研院、資策會、行政院研考會、國家資通安全會報技術服務中心、中華電信、友訊科技、宏達電、趨勢科技、喬鼎科技、中科院、教育部等，並了解他們的檢測需求。舉例說明：我們和工研院資通所達成合作共識，將把本計畫所開發之資訊流動追蹤系統，應用至 ARM CPU 上進行軟體安全分析。此外，我們也和宏達電、友訊科技及中華電信合作，執行軟體安全檢測以及惡意程式行為分析。這些單位在 Access Point/router、智慧型手機也有滲透檢測之需求。在 2011 年，我們針對上述單位的需求來開發適當且客製化的檢測工具。此外，我們也積極地與其他單位聯繫以及交流，來開發出更符合產研單位需求的安全檢測工具。

在 2011 年，本計畫開發了 7 個全新的安全檢測防護工具(請見表 1)。同時，我們也持續把適切的檢測工具轉成線上服務，讓更多人可因此受惠。藉由此平台的建置與檢測工具的開發，我們希望提供政府機關、財團法人及高科技廠商網路安全檢測的服務，並且技轉所開發的檢測工具，以幫助上述單位發現漏洞及弱點。如此一來將可提高產業的經濟效益、提升產品附加價值、節省因網路攻擊或系統弱點所消耗的產值、節省專業檢測人力並且有效減少各種有線、無線網路環境的攻擊。

此外，在 2011 年我們的成果包含技術移轉 1 件、先期技術移轉 1 件、技術服務 1 件、與產學合作達 12 件，總金額超越本計畫書所規畫之 400 萬元。此項成果說明本計畫之建置成果在產業界之可應用性與前瞻性。本計畫之學術以及產業研究均有相當成果。參與本計畫之成員於 2011 年發表於國際重要期刊之論文數共 19 篇(多是 IEEE 以及 ACM 期刊),發表於國際研討會之論文數共 16 篇以及國內研討會之論文共 2 篇。2011 年美國專利獲證 5 件,提出美國以及台灣專利申請各 1 件。詳細列表請參閱（七）計畫成果。

關鍵字: 網路安全、系統安全、軟體安全、惡意程式、社群網路

Abstract

According to the “Policy of Information and Communication Security White Book” and the experience of foreign countries, the TaiWan Information Security Center, namely TWISC, is established under the leadship of Dr. Der-Tsai Lee. In 2005, several research centers including National Taiwan University of Science and Technology (TWISC@NTUST), National Chiao Tung University (TWISC@NCTU) and National Cheng Kung University (TWISC@NCKU) are devoted to security research, secure infrastructure deployment and security knowledge popularization.

The goal of TWISC@NCTU is to develope a heterogeneous network security inspection platform and security tools applied in academic, industry and government in practice. This project can provide the services of heterogeneous network security analysis for any cooperative organization. In 2011, we cooperated with National Security Council, Ministry of Justice, ITRI, III, REDC, ICST, Chunghwa Telecom, D-Link, HTC, Trend Micro, Promise, CSIST, Ministry of Education, etc. The project meets the requirements, which were suggested by the cooperative organizations, to disclosure the vulnerabilities of mobile equipment for security improvement. For example, applying DIFT on ARM-based CPU is one of the work items due to the shared view reached with Information and Communications Research Laboratories, ITRI. Additionally, both testing software security and analyzing behaviors of malware in our research comply with the industry demand by HTC, D-Link and Chunghwa Telecom. These companies are interested in the penetration test of access points, routers, and smartphones, hence our team concentrates on developing the security testing tools meet their requirements, and aggressively contact with the potential cooperative partners for advanced, practical security research.

In 2011, 7 new security testing tools are proposed and implemented with the functionalities required by industry and government, and some functionalities of our project are appropriately turned into on-line services for benefiting the people who are interested in it. With the platform and tools, we anticipate enhancing information security in government sectors, corporations, and hi-tech industries, and we conucted technology transfer with related companies to discover security vulnerabilities in advance. In this way, the quality of network products is increased; the manpower dealing with security threats is reduced; the system vulnerabilities are discovered; finally, the threats in heterogeneous networks are eliminated. Moreover, our team conducted 16 remarkable cooperation projects, which includes industry-university cooperation, technical assistance services, and software licensing, that meet our project goal of 4 million dollars in revenue. This result shows the applicability of our novel techniques. In publication, we have published 7 journal papers, 11 international conference papers, and 1 national conference papers. We also received 3 patents and filed two new patents. Thus, this demonstrated our accomplishments in both academics and industry.

Keywords: Network Security, System Security, Software Security, Malware, Social Network.

一、　背景

異質多網環境中由多種網路(Wired、WiFi、3.5G、WiMAX)構成並存在各式各樣的終端系統，包括個人電腦、筆記型電腦、智慧型手機、PDA、平板電腦(iPad、EeePad）等。以下我們將針對在異質多種網路環境之下，針對系統安全、軟體安全以及人員安全意識來分別介紹本研究計畫相關背景。

■　系統安全

異質多網環境中充滿著各式各樣的終端系統。從個人電腦、筆記型電腦、智慧型手機、電子書、平板電腦(iPad、EeePad）等。這些系統有著各自不同的設計，以及具差異化的產品型態。也因此對於要徹底落實一套有整體性的異質多網系統安全原則（security policy）將是極大的挑戰。再者，使用者在這些終端系統可安裝五花八門的應用程式，並對系統進行各式的調整等。這些安裝與調整的動作都將使系統暴露在”電腦病毒惡意程式”以及“系統漏洞錯誤設定” 等兩大系統安全問題上。

◆　電腦病毒惡意程式

電腦病毒惡意程式經常透過偽裝的方式進行感染散播的目的。這類型的攻擊手法隱藏自己是電腦病毒或是惡意程式的事實，企圖讓使用者執行他所開啟下載的應用程式或是附加檔案，藉此達到感染使用者電腦的目的。此時被感染電腦本身可能被植入後門程式，供惡意攻擊者利用，成為網路上的殭屍電腦，對他人進行 DDOS 等進階攻擊，抑或是單純地被植入病毒，再利用檔案共享、即時通訊、電子郵件等方式伺機傳染下一位受害者。

◆　系統漏洞錯誤設定

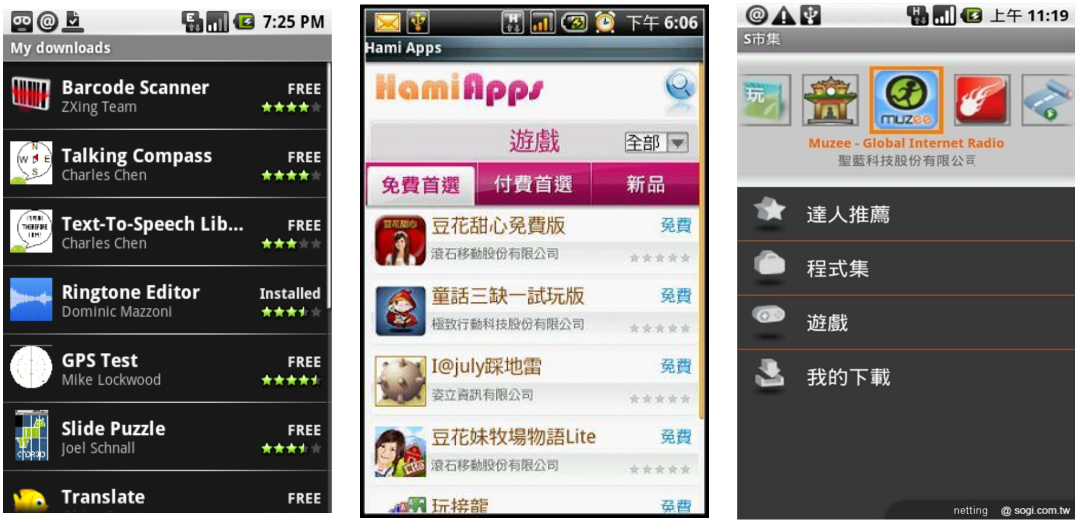
利用使用者系統本身的弱點進行攻擊。攻擊者針對特定的系統或是軟體漏洞，嘗試潛入使用者電腦，跳過身分驗證，取得操作權限，像是開啟 Remote Shell，甚至進一步地獲得管理者權限。這一類的攻擊，攻擊者通常自行尋找系統漏洞，或是利用官方釋出的漏洞修正更新進行逆向工程來發掘漏洞。零時差攻擊便是利用漏洞修正剛釋出，而使用者尚未及時更新時所進行的攻擊行為。攻擊與防護就像時間上的競賽，而不適當的系統設定，亦是攻擊者可利用的弱點，使用者往往做好了各種的安全保護機制，卻因為錯誤的設定，造成系統的漏洞。

以現實情況來說，我們並不能期待異質多網環境中的每位使用者皆具備電腦工程、資訊安全等專業知識背景。這造成了原本一個可以很安全的系統，在落到使用者手中後，很可能就因為使用者不小心裝了個有問題的程式，而在系統上開了個後門。或是有意無意地更動了系統的設定導致原本該有的防護失效等。再者，即便使用者已經知道他的系統並不安全，但由於相關專業知識的缺乏，在多數情況下，使用者也仍舊無法對所面對的系統安全問題進行有效的排除。

過去一年多來Android手機平台迅速竄紅。其成功背後的一個原因是因為Google採取開放平台的策略，也因此各方電信商、手機製造商可以共襄盛舉，一同締造Android平台今日的亮眼成績。然而開放的背後卻也造成一些安全性上的隱憂。比如前些日子，新聞報導了關於一支Android上的應用程式會把個人資訊洩漏到中國大陸。而這背後的原因之一是Google的Android Market本身並沒有採取如Apple App Store一樣，透過一個強制性的中央審核機制來管控所能安裝於手機上的應用程式。因此，無須先通過Google的審查，任何人都可以上傳所開發的應用程式至Android Market讓大家自由下載。這也造成惡意程式相對容易流竄於Android Market上。

台灣的情況又比美國更為複雜。由於我們的電信業者看到Apple App Store的成功背後的巨大商機。也開始仿效Apple App Store模式分別架起各自的Android應用程式市集。這形成目前三強鼎立的情況（見圖1)。可預期的是由於Android平台本身開放的特質，以及台灣目前三頭馬車、缺乏中

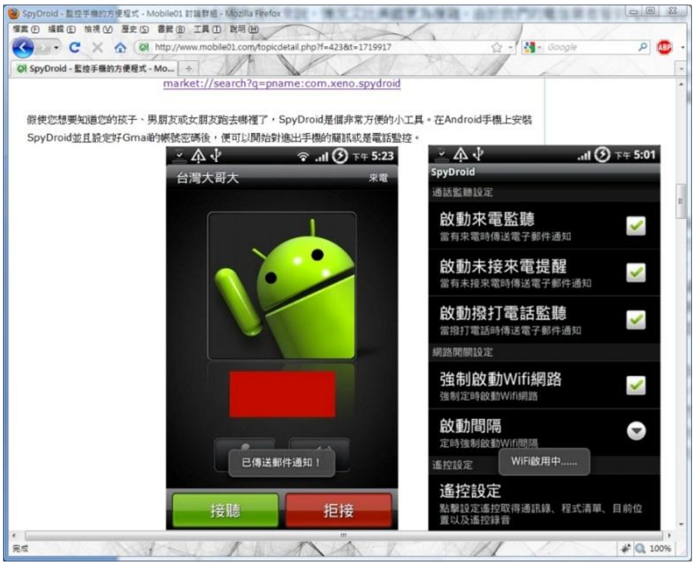
央管控的應用程式市集現況，台灣Android平台上惡意程式的流竄情形恐怕只會更加猖獗。



Google Android Market　　中華電信HamiApps　　遠傳S市集

圖 1. 台灣三個 Android 應用程式市集

在實際生活中，也並不單純只有惡意程式形式的應用程式會對系統安全造成影響。比如諸如SpyDroid（圖2）此類的「善意」監控軟體，如果是在未經使用者許可同意的情況下被安裝於使用者的系統上時，也同樣會對使用者的隱私、系統的安全性造成傷害。由此，我們可以看到系統架構設定對整個異質多網的安全性的影響是一個相當廣泛且具體的問題。


SpyDroid - 監控手機的方便程式 - Mobile01 討論群組 - Mozilla Firefox
檔案(F) 編輯(E) 檢視(V) 歷史(S) 書籤(B) 工具(T) 說明(H)
http://www.mobile01.com/topicdetail.php?f=423&t=1719917
SpyDroid - 監控手機的方便程式 - Mo...
market://search?q=pname:com.xeno.spydroid
台灣大哥大
來電
已傳送郵件通知！
接聽
拒接
SpyDroid
WiFi啟用中......

假使您想要知道您的孩子、男朋友或女朋友跑去哪裡了，SpyDroid是個非常方便的小工具。在Android手機上安裝SpyDroid並且設定好Gmail的帳號密碼後，便可以開始對進出手機的簡訊或是電話監控。

通話監聽設定
- 啟動來電監聽 ☑ 當有來電時傳送電子郵件通知
- 啟動未接來電提醒 ☑ 當有未接來電時傳送電子郵件通知
- 啟動撥打電話監聽 ☑ 當撥打電話時傳送電子郵件通知

網路開關設定
- 強制啟動Wifi網路 ☑ 強制定時啟動Wifi網路
- 啟動間隔 定時強制啟動Wifi間隔

遙控設定
- 遙控設定 點擊設定遙控取得通訊錄、程式清單、目前位置以及遙控錄音

圖2. SpyDroid 手機監控程式

- **軟體安全**

在軟體安全研究領域中，惡意程式分析是極重要的研究議題。九零年代初期，傳統的入侵偵測系統與防毒軟體尚能提供基本的保護能力，然而，近來這些防禦機制已逐漸不敷使用。在惡意程式偵測方面，傳統研究人員大多透過特徵偵測或靜態分析技術，對系統中的檔案或網路中的封包進行

掃瞄與比對，但由於複合型隱匿技術的應用，如：加殼、多型、變型，等技術的精進，惡意程式的特徵可被改變，亦可被複雜化，唯一不變的是惡意程式執行期間對系統的破壞行為。在駭客入侵方面，新型態的駭客攻擊手法不斷翻新，傳統的入侵偵測系統或入侵防禦系統已無法跟上駭客腳步，導致資安事件層出不窮。以跨網站腳本攻擊手法為例，在此手法初現之時，數個國內外大型部落格或相簿網站，如Youtube、Facebook、Flickr與無名等，皆曾遭遇大規模的網路攻擊，為抵禦這些威脅，須依賴大量人力的分析與修補，對比於軟體漏洞與新型惡意軟體的產生速度之快，傳統的防護方法確實不足。以下分別針對複合型的惡意程式隱匿技術和軟體漏洞做進一步的介紹。

◆ 複合型的惡意程式隱匿技術

凡未經電腦使用者同意，逕行於系統內運行對系統進行滲透、破壞或偷竊行為的程式，統稱為惡意程式(Malware)。惡意程式種類繁多，從最古老的電腦病毒，至今日的網路蠕蟲、特洛伊木馬、廣告軟體或間諜軟體等皆屬惡意程式之範疇。值得注意的是，現今的惡意軟體大多結合兩種或多種以上惡意程式之特性進行運作，以達成更強大與多樣化的功能。

在惡意程式的偵測上，傳統的防護方式除了前述之未知惡意程式難以偵測外，還面臨其它問題。大多惡意程式的設計精密且複雜，甚至結合多種隱匿技術，當植入受害者系統中，往往不易被偵測出來。以下舉出惡意程式常見的隱匿技術。

◆ 加殼(Packing)乃是一種最基本的反分析技術。將原本的程式加密或壓縮後，再加上一前導的解壓縮程式即可完成;加殼過的程式,與原本程式特徵不同,且在執行程式時，前導程式會先將程式解殼後再繼續執行。舉例來說，即使是已知的木馬程式，若有攻擊者將其加殼後寄送，即可矇騙許多防護軟體。

◆ 多型(Polymorphism)一開始乃應用在病毒設計上，但現在許多自行散佈的網路蠕蟲也採用此技術，此技術可說是加殼技術的演變；在蠕蟲或病毒散佈自身時，利用不同的加密金鑰使散佈出去的副本皆不相同，以增加防護軟體或進行分析時的困難度。

◆ 變型(Metamorphism)透過 X86 CPU 提供非常強大且複雜的指令集，欲達成同一行為的程式可有十分多樣化的程式寫法。如：mov eax, 0h 可寫成 xor eax, eax 或是 sub eax,eax 等，因此將這些寫法建立成自動化的變體引擎，將可做到百分之百的變型效果，且毋須前導解殼行為，大幅降低被偵測軟體發現的可能。

◆ 核心等級匿蹤技術(Kernel Level Rootkit)乃近年來最新型態的惡意程式設計技術。其作用在於隱藏惡意程式的存在，包括硬碟中的惡意程式檔案本體、記憶體中運行的程序、傳送至網路上的封包，亦或是聽取網路連線的連接埠等，皆可被完全隱藏，也因此許多惡意程式一但侵入系統後便難以移除。這種 Rootkit 的作法藉由驅動程式的形式入侵至作業系統的核心中，可取得與作業系統同樣的最高執行權限，再透過修改某些系統重要資訊的方法，達成隱藏重要資訊並攔截重要系統訊息的目的。

多變的惡意程式撰寫技術令現有的分析技術失去了效用。其中兩點重要的觀察值得研究：其一，惡意程式可欺騙使用靜態特徵碼比對的傳統防毒軟體。其次，惡意程式開始侵入作業系統核心，可奪取最高執行權限，因此處在與惡意程式同樣權限的防毒軟體將會受到惡意程式的影響，所以我們需要更高的層級來對這些 Rootkit 技術進行分析。

◆ 軟體安全性漏洞

軟體安全一直是很重要的議題。即使是來自全球各地的無數專業程式設計者所開發出來的 Windows 系列作業系統，尚有許多安全性漏洞，也時時刻刻都有新的漏洞被發現，以至於相關的 security patches，在 Windows 系列作業系統問世以來，依然需要每個星期持續更新。圖 3

顯示National Vulnerability Database最近十年(2001年1月 ~2010年9月)的Vulnerability統計，在2010年1月至9月就有多達3568個漏洞被發佈。

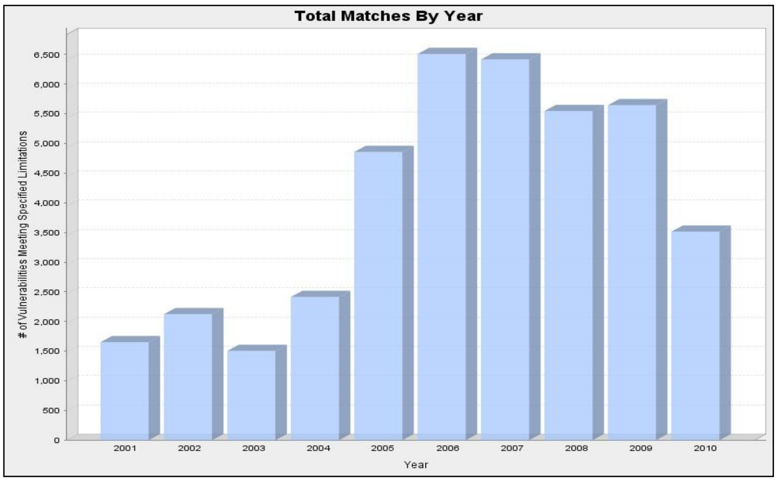


圖 3. Vulnerability statistics (2001年1月至2010年9月) from National Vulnerability Database

然而，程式漏洞有成千上萬種，在軟體方面的安全性漏洞大致可分為以下類型：(1) Memory safety violations、(2) Input validation errors、(3) Race conditions、(4) Privilege-confusion bugs、(5) Privilege escalation、(6) User interface failures。例如，在Memory safety violations中，最常見的攻擊方式為Buffer overflows。駭客可以利用這個漏洞，使得buffer的寫入超出原本的長度限制，輕易的控制電腦中的記憶體位置及資料，進而獲得作業系統的root權限。Input validation errors類別中包含了在Web應用程式中很常見的SQL injection攻擊方式，在輸入的字串中夾帶惡意的SQL指令，讓資料庫伺服器誤認為是正常的SQL指令而執行，造成資料庫資料的外洩。以上各式各樣的漏洞，大部份來自於programmer在撰寫程式碼時的不注意、疏忽所產生的，也對整個系統造成了安全性的威脅。

■ 人員安全意識

隨著網際網路的蓬勃發展以及虛擬網路社群(Virtual Community)的興起。「社群網路」係將實體社會中的社區、團體概念延伸到網路上，社群網站也適度滿足了人們主宰及炫耀的心理，因為任何人都可以成立網路社團，而不需要任何金費或人脈，便可擁有版主、會長的頭銜。然而，隨著社群網站使用者的成長與特殊社群的激增及行動通訊的便利，這些因素皆增強了社群網路的影響力，同時間線上威脅與弱點因此也不斷升級。面對熱門社群網站上逐漸頻繁的網路入侵與問題，證實了使用者們必須建立對網路威脅的警覺心並養成線上系統防護的習慣。雖然大多數社群網站使用者皆遭受到線上危機的威脅，但只有不到三分之一的使用者會主動採取防護措施。根據「諾頓家庭防護網報告」指出，使用者保護個人電腦安全的意識與常識嚴重不足，近八成以上的使用者沒有加裝防毒軟體或更改密碼的習慣；更有高達82%的使用者會相信不實的網路廣告；此外，近四成的使用者曾下載帶有病毒的檔案到個人電腦或家庭用電腦。反映出台灣使用者對於網路虛擬世界的過分信任。因此，針對培養人員安全意識及增進線上系統防護的警覺，我們分別探討使用者於社群網站中常遭受的網路攻擊：

◆ 網路釣魚程式

網路釣魚者（Fisher）傳送包含釣魚網站連結的電子郵件給使用者，欺騙他們到釣魚網站輸入個人或財務資料如身份證號碼、銀行帳號及信用卡號碼等等，然後盜用這些資料謀取利益。

網路釣魚駭客也可能直接攻擊社群網路服務。駭客們可在熱門使用者的留言板上以其朋友的姓名貼上偽造的網路服務登入網頁，當其他使用者瀏覽該留言板並連結到釣魚登入網頁時，他們可能以為是偶然地被服務系統登出（例如 session 時間超出），立即輸入帳號密碼欲重新登入。得到某使用者的登入資料後，網路釣魚駭客就可以對其在社群網路服務的朋友使用更多的詐騙手法。

◆ 垃圾郵件

一般來說，使用者會把好友的電子郵件地址設為白名單（white list， 例如聯絡人或垃圾郵件過濾器的例外名單），所以從朋友郵件地址寄來的郵件不會被判定為垃圾郵件。如果垃圾郵件發送者利用使用者朋友的電子郵件地址來偽造郵件，所送出的垃圾郵件就可能獲得更多的信任，甚至可以騙過垃圾郵件過濾器。此外，如果垃圾郵件發送者在郵件的主題或內容中加入使用者好友的姓名或暱稱，即使該郵件已被判定為垃圾郵件，使用者仍可能會開啟它。在此情況下，垃圾郵件即使已被成功阻擋，也能對使用者造成困擾。目前類似案例數量已持續成長，成為反垃圾郵件研究者的新挑戰。

◆ 惡意攻擊

網路攻擊者透過入侵合法網站偷偷地影響網路使用者的情況越演越烈，偷渡式下載 (Drive-by-Downloads) 蔚為風潮，網路攻擊者越來越朝向直接鎖定最終使用者/一般使用者，然後企圖誘騙一般使用者下載惡意軟體或是誘使其在認為一切安全的情況下，無意地洩漏敏感資訊。社交工程之建置與使用者電腦上所採用的作業系統和網路瀏覽器並沒有相關性，因此攻擊者鎖定的目標所針對的是實際的使用者，而非機器本身的漏洞。

二、 研究目的與成效

依據行政院『我國資通安全政策白皮書』、國外經驗與借鏡，設置國家級之「資通安全研究與教學中心(TaiWan Information Security Center, TWISC)」，在中央研究院李德財院士的領導下，於 94 年成立台灣科技大學資通安全研究與教學中心(TWISC@NTUST)、交通大學資通安全研究與教學中心(TWISC@NCTU)、與成功大學資通安全研究與教學中心(TWISC@NCKU)，分別從事其專長領域研究、技術建置與推廣。

TWISC@NCTU 在本計畫主要的目的在於建置一個異質多網安全檢測平台、開發與建置產官學研所需的安全檢測工具，以及提供政府機關、產業界或是財團法人異質多網安全檢測的服務。希望藉由我們所開發各種安全檢測工具來發現異質多網與行動設備潛在的安全問題，讓管理人員以及一般使用者可提早修補漏洞、改善問題以提高行動設備安全性。為了在 2011 年開發適合產官學研的安全檢測工具，我們已經與多個政府機關、法人以及產業界建立合作關係，包括：總統府國安會、法務部調查局、工研院、資策會、行政院研考會、國家資通安全會報技術服務中心、中華電信、友訊科技、中科院、教育部等，並了解他們的檢測需求。舉例說明：我們和工研院資通所達成合作共識，將把本計畫所開發之資訊流動追蹤系統，應用至 ARM CPU 上進行軟體安全分析。此外，我們也和友訊科技及中華電信合作，將做軟體安全檢測以及惡意程式行為分析。這些單位在 Access Point/router、智慧型手機也有滲透檢測之需求。除了行動核心網路拓樸探索工具之外，中華電信也對於本計畫於 2011 年所開發的社群網路可疑連結隱藏內容發掘系統也有合作的意願。在 2011 年，我們針對上述單位的需求來開發適當且客製化的檢測工具。此外，我們也積極地與其他單位聯繫以及交流，來開發出更符合產研單位需求的安全檢測工具。

在 2011 年，本計畫開發了 7 個全新的安全檢測防護工具(請見表 1)。同時，我們也將持續把更多適切的檢測工具轉成線上服務，讓更多人可因此受惠。藉由此平台的建置與檢測工具的開發，我們希望提供政府機關、財團法人及高科技廠商無線網路安全檢測的服務，並且技轉所開發的檢測工具，以幫助上述單位發現漏洞及弱點。如此一來將可提高產業的經濟效益、提升無線產品附加價值、節省因網路攻擊或系統弱點所消耗的產值、節省專業檢測人力並且有效減少各種有線、無線網路環境的攻擊。

表 1. 檢測工具開發、升級以及維護之清單

於 2011 年全新開發之檢測工具	線上 Windows 與 Linux 系統組態安全檢測系統
	作業系統 DNS 快取毒害監測與防護工具
	線上即時軟體行為分析檢測工具
	社群網路可疑連結隱藏內容發掘系統
	行動核心網路拓樸探索工具
	行動平台漏洞模擬系統
	Agent-based 合作式滲透測試系統

此外，在 2011 年本團隊在技術移轉、軟體授權、技術服務與產學合作總金額超出原本計畫書所規畫之 400 萬元。此項成果說明本計畫之建置成果在產業界之可應用性與前瞻性。在本年度中我們進行技術移轉“建構於行動裝置 ARM CPU 上之污染分析系統”至工研院，並接受友訊科技委託，進行多年期“委託 Open D-Link Routers Forum 建置、維護與測試技術與諮詢服務”的技術服務，在表 2 中列出今年度產學合作之對象與計畫名稱，詳細 KPI 請見第肆章。

表 2. 2011 產學合作對象與計畫名稱

合作對象	方式	計畫名稱
工研院	技術移轉	建構於行動裝置 ARM　CPU 上之污染分析系統
宏達電子(HTC)	先期技轉以及產學合作	雲端惡意程式鑑識與行動平台安全
D-Link 友訊科技	技術服務	委託 Open D-Link Routers Forum 建置、維護與測試技術與諮詢服務Ⅱ
法務部調查局	產學合作	法務部調查局惡意程式自動檢測技術支援系統委託研究採購案
教育部	產學合作	DNSSEC 推動先期型計畫
中華電信	產學合作	動態惡意程式行為側錄與污染分析
中華電信	產學合作	行動平台資通訊安全問題的研究(二)
中華電信	產學合作	基於虛擬網路技術適用於異質網路之資源分配最佳化
喬鼎科技	產學合作	前瞻性檔案完整性驗證與可疑嵌入碼檢測平台
工業技術研究院	產學合作	雲端行動的安全及時分析可行性評估先期探討
工業技術研究院	產學合作	智慧終端技術研究
工業技術研究院	產學合作	行動終端軟體品質技術研究
中華電信	產學合作	雙階層式全系統汙染鑑識分析
中華電信	產學合作	行動平台資通訊安全問題的研究(三)
教育部	產學合作	DNSSEC 網域名稱安全架構建置與推廣計畫
趨勢科技	產學合作	Technology Transfer on Network Threat Detection using Security Log Correlation

三、 國內外相關研究

以下將針對各研究範疇，介紹目前國內外相關研究。

■ 系統安全

弱點掃描與滲透測試是常見用來評估系統的安全程度的方式。常見的弱點掃描軟體有Nessus [1]、Acunetix Web Vulnerability Scanner [2]等。在弱點攻擊方面的工具則有SAINTexploit [3]、Metasploit [4]、CORE IMPACT[5]等。

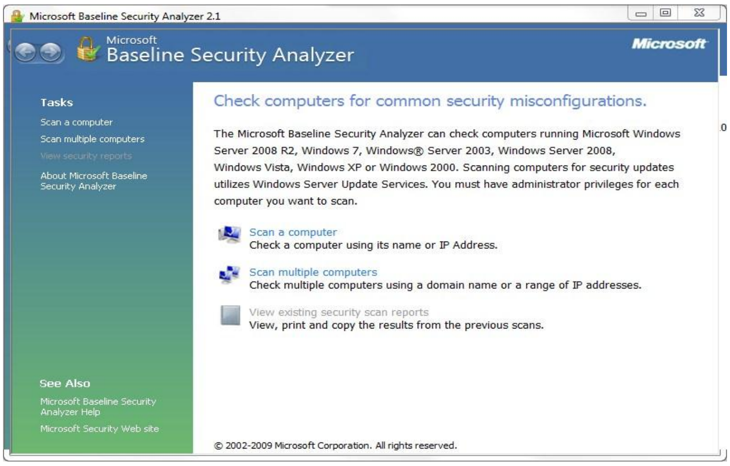
Metasploit是一套已經發展多年的滲透工具，提供攻擊函式庫及攻擊封包，可以滲透系統，取得系統權限，但Metasploit並沒有實現滲透測試自動化，測試者需先藉由Nmap和Nessus等埠掃描和弱點偵測軟體找出連接埠和弱點，再自行由Metasploit攻擊函式庫裡找出適用的攻擊程式。CORE IMPACT是一套商用滲透測試工具，它提供一個具擴充性的滲透測試框架。CORE IMPACT的主要滲透測試手法是利用受害主機的弱點滲入主機，接著植入一個代理程式，代理程式除了會掃描其他自身的弱點和發動網路攻擊，它也會啟動自動掃描，尋找網路系統中的下一個可能受害主機。另外，國內外學者對於系統安全、弱點掃描、滲透測試也有相關的研究成果 [6,7,8,9,10,11] 。滲透測試的過程由一連串的測試步驟、搭配不同的軟體與滲透路徑、分析方法組合而成，以對大型企業的實體網路進行滲透測試為例，滲透測試者可利用常見的SQL injection攻擊或是DoS攻擊等手法對企業的對外網路服務，進行攻擊演練，找出潛在漏洞和攻擊路徑。我們以資訊量最少的黑箱測試為例，將滲透測試可分為探測、弱點掃描與弱點攻擊、結果分析等四大步驟[15]。

- ◆ 探測：探測的目的在於收集以及偵查受測者的環境與背景，例如網路拓墣、受測主機的作業系統(OS)等等。常見的探測軟體為通訊埠掃描軟體 Nmap [16]，檢測者能透過不同的參數設定，得到不同的受測者資訊，諸如受測者的作業系統、開啟的通訊埠、防火牆的通訊埠過濾規則、正在提供的網路服務等。
- ◆ 弱點掃描：在了解受測者的背景之後，滲透測試者會根據得到的受測者背景資訊，諸如開啟的通訊埠與使用中的網路服務、使用的軟體版本等資訊，進行更進一步的弱點掃描來確認受測者所開啟的服務中是否存在進行攻擊的管道。常見的弱點掃描軟體有 Nessus、SAINTexploit、Acunetix Web Vulnerability Scanner [2]等。Nessus 為一廣泛使用的弱點掃描軟體，它能針對目標主機或網路安全弱點產生評估報告，提供滲透測試者目標主機的安全弱點與安全漏洞等訊息，並提供相關之說明連結等。SAINTexploit 是 SAINT 公司所開發的一套網路系統潛在安全漏洞的搜尋工具，它可以用於找出攻擊者可能入侵網路的漏洞並且計算網路風險值。
- ◆ 弱點攻擊：此步驟對掃瞄出來的弱點進行滲透攻擊，嘗試進入目標網路或是取得目標主機的 root 權限。弱點攻擊常用的工具有 Metasploit、CORE IMPACT 等。Metasploit 是一套已經發展多年的滲透工具，提供完整且不斷更新的攻擊函式庫及攻擊封包，滲透測試者可藉由 Nmap 和 Nessus 等埠掃描和弱點偵測軟體找出的連接埠和弱點，從攻擊函式庫中選取適用的攻擊程式進行攻擊。CORE IMPACT 是一套商用滲透測試工具，它提供一個具擴充性的滲透測試框架。CORE IMPACT 通常先針對一台主機的弱點進行滲透攻擊，接著在被破解的主機中植入一個代理程式。代理程式除了會持續掃描被破解的主機中是否存在其他的弱點外，也會啟動自動掃描，嘗試從目標網路的內部尋找網路系統中的下一個可能受害主機，並發動滲透攻擊。

- 結果分析：完成上述的步驟後，測試者可依據掃描和攻擊測試結果，完成一份全面性的整合測試報告，說明目標網路或系統中存在的弱點、攻擊者可能的攻擊路徑與手段，以及攻擊成功後，受測主機可能遭受的損害。測試報告也會包含對受測者的補強建議，加強受測機器的安全性和強健度，使有心人士無法針對漏洞有機可乘。

雖然目前的滲透測試多半是針對實體網路主機進行安全檢測，但是隨著行動網路的蓬勃發展，透過WiFi或是3.5G上網的行動裝置也逐漸成為駭客潛在的攻擊目標。

在系統組態設定檢測方面，Alfaro等人與Oliveira 等人 [12,13]探討了如何偵測防火牆上不當的組態設定。Oliveira等人[13]偵測諸如冗餘的防火牆規則、shadowing anomalies等會造成防火牆不正常運作的一些組態設定上的問題。另一方面，像微軟的Baseline Security Analyzer (圖4) 可以用來偵測Windows系統本身的一些組態問題(如圖5中 Incomplete Updates, Password Expiration, File System, 等)。以這些例子來說，他們多半針對特定的系統（如防火牆或Windows），且多半只關注該系統本體。以微軟Baseline Security Analyzer來說，其並沒有考量到安裝於Windows上面的應用程式相關的組態設定問題。在偵測技巧上，這些現存系統多半可以透過對個別系統所特別打造的偵測法則來檢查系統組態是否有問題。好處是或可很迅速準確地抓出所關心的問題，壞處則是無法應用到不同的系統甚至是同系列系統的新版本。


Microsoft Baseline Security Analyzer 2.1
Microsoft
Baseline Security Analyzer
Microsoft
Tasks
Scan a computer
Scan multiple computers
View security reports
About Microsoft Baseline Security Analyzer
See Also
Microsoft Baseline Security Analyzer Help
Microsoft Security Web site

Check computers for common security misconfigurations.

The Microsoft Baseline Security Analyzer can check computers running Microsoft Windows Server 2008 R2, Windows 7, Windows® Server 2003, Windows Server 2008, Windows Vista, Windows XP or Windows 2000. Scanning computers for security updates utilizes Windows Server Update Services. You must have administrator privileges for each computer you want to scan.

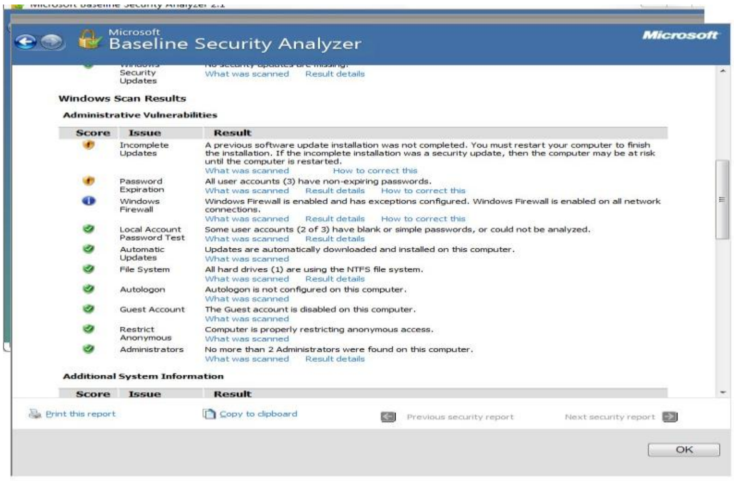
Scan a computer
Check a computer using its name or IP Address.

Scan multiple computers
Check multiple computers using a domain name or a range of IP addresses.

View existing security scan reports
View, print and copy the results from the previous scans.

© 2002-2009 Microsoft Corporation. All rights reserved.

圖 4. Microsoft Baseline Security Analyzer



	Security Updates	What was scanned Result details

Windows Scan Results

Administrative Vulnerabilities

Score	Issue	Result
!	Incomplete Updates	A previous software update installation was not completed. You must restart your computer to finish the installation. If the incomplete installation was a security update, then the computer may be at risk until the computer is restarted. What was scanned How to correct this
!	Password Expiration	All user accounts (3) have non-expiring passwords. What was scanned Result details How to correct this
i	Windows Firewall	Windows Firewall is enabled and has exceptions configured. Windows Firewall is enabled on all network connections. What was scanned Result details How to correct this
✓	Local Account Password Test	Some user accounts (2 of 3) have blank or simple passwords, or could not be analyzed. What was scanned Result details
✓	Automatic Updates	Updates are automatically downloaded and installed on this computer. What was scanned
✓	File System	All hard drives (1) are using the NTFS file system. What was scanned Result details
✓	Autologon	Autologon is not configured on this computer. What was scanned
✓	Guest Account	The Guest account is disabled on this computer. What was scanned
✓	Restrict Anonymous	Computer is properly restricting anonymous access. What was scanned
✓	Administrators	No more than 2 Administrators were found on this computer. What was scanned Result details

Additional System Information

Score	Issue	Result

圖 5. MBSA Scan Result

■ **軟體安全**

軟體錯誤的產生原因大多來自於程式設計者對於程式流程的失控，且發現錯誤發生時往往不易經由人工方式分析出真正的原因，因此我們需要工具程式來協助分析錯誤點。另外假若軟體錯誤可被利用，這些錯誤就很有可能被轉化為安全弱點。早期的軟體安全測試方式是經由隨機方式產生的測試資料輸入給軟體，並分析軟體的執行狀況。近幾年來軟體檢測的方式逐漸有系統化，國內外研究範疇大致上可分為兩大類 :

◆ 靜態分析(Static Analysis) - 是對軟體程式碼作靜態程式碼分析(static code analysis)，檢驗程式碼中是否有錯誤或可能被攻擊的弱點存在，而不需要測試人員的幫助就能作自動化的測試。然而，靜態測試有其無法避免的缺點，就是誤判率(false positive, or false alarm)過高的問題。只是靜態分析程式碼而沒有實際執行，無法保證找到的程式錯誤是真正的錯誤。所以靜態分析的結果需要靠測試人員進一步檢查才能確定是誤判或是真實的錯誤，但誤判的比例通常不小，使得在測試軟體上的人力花費仍是不少，相關研究工具有 Findbugs、PMD 等等。

◆ 動態測試 - 與靜態測試最大的不同在於，實際執行來確定是否會引發程式的錯誤。因為實際執行的關係，只要是在執行時期碰到的錯誤很明顯都是真正的錯誤，不會再有誤判的問題。不過要實際執行一個程式會遇到另一個關鍵的問題，如何提供這程式所需要的輸入，例如函式的參數或標準輸入等。最簡單的方式是亂數產生輸入給程式的測試資料，這樣的工具稱 fuzzer。因為是亂數產生程式的輸入，fuzzer 很容易產生一堆輸入資料給受測程式，有一定的機率讓程式行為異常。Fuzzer 把受測程式視為黑盒子，只了解程式的輸入輸出，對其內部結構完全沒有分析，所以無法測試的範圍無法涵蓋所有程式的路徑，沒辦法測到需要特殊條件的路徑或結構，相關研究有 jCute[19, 20]、java path finder[18]等等。

如今，一種結合靜態測試與動態測試的方法被提出，叫做concolic testing[21,22]，一方面利用symbolic execution[23]對所有被程式輸入影響到的變數作符號分析(symbolic analysis)，以解決產生輸入的問題；另一方面用concrete execution確定目前輸入所引導的程式執行路徑，來避免誤判率高的缺點。Conclic testing將靜態與動態分析結合以達到優缺點互補的效果，在近幾年學術界有許多相關的研究顯示這個方法比起傳統的測試方法有很大的改善，而且被應用在系統核心的測試，以及多線程(multi-thread)的軟體測試上。

另一方面，惡意攻擊者會蓄意利用軟體設計錯誤的漏洞設計出相對應的惡意軟體進行系統攻擊。現行對於惡意軟體檢測的防毒軟體(如: kaspersky[14]，Norton[17])多是利用病毒特徵碼進行惡意軟體判定。惡意軟體為了隱藏自己的存在，不被防毒軟體偵測，因此常透過加殼的方式來打亂程式特徵碼，逃過bit/byte特徵碼比對追蹤。只有在程式執行的過程中，惡意軟體才會依序將程式解譯出來並寫入記憶體執行。因此在執行前，是無法透過特徵碼檢驗的方式發現此類惡意程式。目前已有許多研究希望能夠以自動化的方式解譯惡意程序正確執行順序的方法。不過目前為止，除了特定已知的加殼方式之外，無一方法可針對特殊加殼的方式進行解譯。由於執行任何程式之前，必須先將該程式載到記憶體中才能執行，惡意程式當然也不例外。因此在Renovo[41]的方法中提出，使用模擬環境，讓目標程式在模擬系統上執行，並監控所有進行記憶體寫入與程式執行流程變更的指令。此外，開闢一塊虛擬的記憶體區塊，標記程式執行過程新寫入記憶體的程式碼，最後將這些程式碼依序Dump出來，以取回程式碼執行的正確順序，便能對程式碼進行特徵碼比對。

■ **人員安全意識**

許多的企業花費高昂的成本添購資訊安全設備，但資安事件未曾在企業網路銷聲匿跡?細究其原因，不難發現，太多的企業把資訊安全當作技術問題來處理，而忽略了管理的重要。許多企業儘管擁有了先進而昂貴的資訊安全設備，但「政策」或「人」往往無法配合，產生了很大問題。比如系統缺乏妥善的監控與持續的警覺性、錯誤的設定等等。因此，最好的資安設備未必是最安全的設備。許多安全問題不是因為產品的功能不佳造成；即使企業上了層層關卡，卻可能因為被破解的管理者密碼、help desk維修後未關閉的開放權限、輕易分享所有人的網路資料夾…等管理盲點，為駭客及病毒開啟了許多後門。因此，我們不難發現，企業資訊安全是「人員管理問題」，而非單純的「技術問題」。唯有妥善的管理，才能降低風險，避免不必要的損失，並能持續企業的營運，積極地掌握每一個商機。針對人員安全意識研究範疇探討國內外相關研究，我們分為下列三點說明：

◆ 網路釣魚程式 - 現今網路科技發展蓬勃，使用者非常容易遭受到網路釣魚的攻擊，因此教育使用者網路釣魚攻擊的知識觀念相當重要,使用者了解的越多越能減少被成功攻擊。Anti-Phishing Phil[24]藉由自行設計的一個網路釣魚遊戲，讓使用者在遊戲中學習網路釣魚的觀念與特徵，以期往後在使用網路時不會掉入釣魚者的陷阱。Kumaraguru et al.[25]藉由如何設計教材的研究分析，讓使用者學習過網路釣魚的教材後，能確實記憶此網路釣魚觀念並能轉換知識，使用者能融會貫通識破相關網路釣魚攻擊。除了教育使用者具備相關網路釣魚之知識外，亦有些各類型偵測網路釣魚的研究。釣魚郵件暗藏惡意超連結，藉由分析 URL 的結構來偵測網路釣魚攻擊，釣魚者會利用 URL Obfuscation[26]的方式欺騙受害者；Blacklist Generator[27]建議一個產生釣魚網站黑名單的架構，如何能夠提供一個快速更新的黑名單；另外網路釣魚攻擊也有利用 Pharming Attack 的方式，竄改 DNS 紀錄來達成攻擊目的，Dynamic Pharming Attack[28]是種新型態的攻擊方式。

◆ 垃圾郵件 - 大量寄發垃圾郵件的行為會造成頻寬、儲存設備、時間、生產力等的資源濫用，導致嚴重的經濟損耗。垃圾郵件製造者(Spammer)主要是利用僵屍電腦(Zombie computer)構成的殭屍網路(Botnet)[29]來發送垃圾郵件。所謂的僵屍電腦指的是遭到入侵而被利用來進行犯罪行為的電腦，透過病毒郵件傳播又會產生更多的殭屍電腦，如此惡性循環，構成一個龐大的殭屍網路。要完全抑止垃圾郵件來源發信幾乎是不可能，因此，對於垃圾郵件的防治，如何有效的阻擋、分析以及分類就顯的更為重要。一般垃圾郵件的辨識機制主要分為兩類，身份驗證和內容分析。身分驗證是拿已知的各種資訊來驗證信件來源是否符合系統自訂的條件，符合則通過，否則阻擋，相關技術如黑名單、白名單、灰名單、SPF 等。由於身分驗證機制只是做簡單的資訊比對，處理時間較短，但容易因為資訊不足、過久未更新名單以及寄件方伺服器未支援驗證機制等因素，造成 false positive 或 false negative[30]。false positive 是指正當訊息因系統誤判而被分類成垃圾郵件, false negative 是指因資訊不足，造成垃圾郵件沒有被正確分類的情況。

內容分析，是透過各種特徵以及規則，分析郵件本文後再評估目標為垃圾郵件的可能性，一套完善的內容分析機制能夠有效的鑑別垃圾郵件，如 Spamassassin[31]。內容分析雖然對垃圾郵件具有高度辨識率，但由於分析規則繁瑣，處理時間較長，因此一般伺服器管理者會搭配身分驗證機制一起使用，以提高分析系統的整體效能。因考慮到內容分析機制會接觸郵件本文以及運算複雜度較高等因素,部分學者也著手研究以郵件日誌為基礎的分析技術，希望藉由日誌檔上提供的郵件收送資訊以及系統訊息，達到近似內容分析機制的辨識效能，一方面可減低分析郵件的時間成本，另一方面可避免接觸他人隱私。

黑名單驗證是屬於早期的郵件驗證機制，透過將已知垃圾郵件發佈者的 IP 或 Domain 建立成一組名單[32]，往後外界來信送達時，該機制會先將寄件來源與此名單比對，若該來源出現在名單上則阻擋信件。但由於 SMTP 上的漏洞使信件來源可以偽造，利用 Open relay 的郵件主機將信件轉送亦可隱藏身分，並且隨著名單內容的增加，龐大的查詢也可能給系統帶來不小的負擔，因此傳統的黑名單機制已不適用在目前的電子郵件通訊上。目前取而代之的是由傳統黑名單機制衍申出來的「及時性黑名單」 (Real-Time Blackhole Lists，RBL)[33]。郵件伺服器可透過即時查詢 RBL 來判斷是否阻擋信件，可省去傳統黑名單查詢造成的負擔，亦可獲得較全面的資訊。但也由於此服務的及時性，資料庫的定期更新及維護也決定了過濾惡意來源的效能，若選擇過久未更新的資料庫可能會導致嚴重的 false positive 或 false negative，因此選擇有信譽的 RBL 作為黑名單查詢是非常重要的一環，如「ORBD」[34]是目前外界公認較準確且免費的 RBL。在本研究嘗試利用離線型分散式日誌分析及位址檢查機制製做 RBL，希望能提升郵件的分析速度及效能。

「貝氏分類法」[35]是利用「貝氏定理」發明的分類法則。「貝氏定理」是將事前機率與條件機率結合，最後導出事後機率的過程。此技術先將信件分割成 n 個斷字(Token)，再透過演算法統計個別斷字的機率，進而推算出為垃圾信的可能性。「貝氏定理」主要仰賴過往累積的數據來判斷事件發生的機率。因此對於剛建置的「貝氏過濾法資料庫」，如能事先各提供 1000 封垃圾信及正常信件，絕對有助於信件辨別的訓練。首先，分別將 1000 封的正常信與垃圾信放入資料庫，系統會將這 2000 封信的內容切成斷字，給予不同的機率。當下次接收到新進郵件，一樣會把信件分解成斷字，與訓練過的「貝氏過濾法資料庫」做比對，依據過往的經驗做分析，能精確評估此郵件為垃圾信件的機率。

◆ 惡意攻擊 - 根據賽門鐵克(Symantec)公司調查統計，光 2009 年企業遭受資訊安全問題平均每家損失 200 萬美元 [36]。由此可見確保資料安全與伺服器正常運作一直以來都是網管人員必須面臨的考驗，如何兼顧高可用性(High-Availability)、安全性(Security)與低延遲(low-latency)是資訊安全最高深的學問 [37,38]。

網路應用程式與動態網頁技術讓網頁從過去簡單陽春的表現形式，轉變為具聲光動畫效果、功能創新和多樣(如社群網路、網路拍賣交易、網頁遊戲等) 的網頁。不幸的是，也讓一些有心人士利用自己架的伺服器網頁或是公開允許使用者輸入的網站(如部落格、討論區等) 來撰寫不恰當的客戶端指令碼程式(像是一直開啟視窗讓使用者系統當機、強制轉址到其他網站)，或是輸入一些惡意的連結引誘其他使用者點擊等等。亦或是利用網路應用程式開發者的程式漏洞，譬如將使用者輸入的資料字串直接傳回資料庫，則會造成 SQL Injection 漏洞的產生，有心人士則會利用這漏洞注入一些惡意程式碼或是不需帳號密碼便可進入帳號模式。亦或是利用網頁瀏覽器漏洞來進行攻擊或是犯罪。

駭客的攻擊目標大多為使用者在各公司申請過的帳號與密碼、網路銀行之密碼、信用卡或是提款卡之編號與密碼，也有為了竊取重要個人資料、或是公司資料庫中所存放的客戶資訊等。而近年來駭客的攻擊手法更新，以網路應用程式當作媒介，注入惡意程式碼讓使用者自動轉址、開啟惡意網站,或是使用隱匿強迫下載(Drive-By Download) 的方式而下載惡

意程式(如木馬程式、鍵盤側錄、病毒、蠕蟲等)。

Mihai Christodorescu 與 Somesh Jha Christodorescu and Jha [39] 於 2004 年針對當時商業防毒軟體進行了惡意軟體所常用之混淆技術的耐受性進行檢測，結果顯示，各家的防毒軟體誤判率(False Negative Rate)平均從四成到八成皆有，甚至還有出現完全誤判的情形。而真正能辨識出是否為惡意網頁或是惡意程式碼，須經由訓練有素或是經驗老到的資安人員以人工的方式來做判斷。惡意程式執行(Malicious File Execution)是 Web 應用程式引入來自外部的惡意檔案並執行檔案內容，大多發生在 PHP 程式語言上。一般在做 Web 應用程式的弱點掃瞄時，大部份的工具很難檢測出哪些參數會去讀取檔案[40]。

四、 重要參考文獻

[1]. Nessus, http://www.nessus.org

[2]. Acunetix Web Vulnerability Scanner, http://www.acunetix.com/vulnerability-scanner/

[3]. SAINTexploit, http://www.saintcorporation.com/products/penetration_testing/saint_exploit.html/

[4]. The Metasploit Project, http://www.metasploit.com

[5]. CoreImpact, http://www.coresecurity.com/content/core-impact-overview

[6]. N. Provos. 2003, *Improving host security with system call policies*. In Proceedings of the 12th Conference on USENIX Security Symposium - Volume 12. USENIX Security Symposium. USENIX Association, Berkeley, CA, pp. 18-18.

[7]. D. Mazières, M. Kaminsky, M. F. Kaashoek, and E. Witchel. 1999, *Separating key management from file system security*. In Proceedings of the ACM SIGOPS - Volume 33, pp. 124-139. Available: http://doi.acm.org/10.1145/319344.319160

[8]. J. P. McDermott. 2000, *Attack net penetration testing*. In Proceedings of the Workshop on New Security Paradigms. NSPW '00. ACM, New York, NY, pp. 15-21. Available: http://doi.acm.org/10.1145/366173.366183

[9]. 柯鈞凱、楊中皇。結合弱點掃描和滲透測試之自動化Web安全檢測系統設計與實現。第二十屆資訊安全會議（CISC 2010）。

[10]. 鐘崇斌。系統/網路安全弱點掃描。NSC91-2622-E009-027-CC3。

[11]. 鐘崇斌。系統弱點掃描之代理程式設計。NSC92-2622-E009-014-CC3。

[12]. J. Alfaro, et al. 2008, *Complete analysis of configuration rules to guarantee reliable network security policies*. In Proceedings of the International Journal of Information Security - Volume 7, pp. 103-122.

[13]. R. M. Oliveira, et al. 2009, *Automatic Detection of Firewall Misconfigurations using Firewall and Network Routing Policies*. In Proceedings of the IEEE DSN Workshop on Proactive Failure Avoidance, Recovery, and Maintenance, Lisbon, Portugal.

[14]. 卡巴斯基實驗室, http://www.kaspersky.com.tw/

[15]. Russell Dean Vines Penetration testing tutorial, http://searchsecurity.techtarget.com.au/articles/23373-Penetration-testing-tutorial-Day-One-The-basics

[16]. Nmap, http://www.nmap.org

[17]. 諾頓(賽門鐵克), http://tw.norton.com/

[18]. W. Visser, C. Pasareanu, S. Khurshid. 2004, *Test Input Generation with Java PathFinder*. In Proceedings of the ISSTA. Boston, MA.

[19]. S. KOUSHIK, and A. GUL. *jCUTE : Automated Testing of Multithreaded Programs Using Race-Detection and Flipping*. Submitted for Publication.

[20]. S. KOUSHIK, and A. GUL. 2006, *CUTE and jCUTE : Concolic Unit Testing and Explicit Path Model-Checking Tools*. In Proceedings of the 18th International Conference on Computer Aided Verification (CAV'06), Lecture Notes in Computer Science, Seattle, Washington, USA.

[21]. S. Koushik. 2007, *Concolic Testing*. In Proceedings of the EECS Department, UC Berkeley, CA, USA. ASE'07.

[22]. S. KOUSHIK, and A. GUL. *Concolic Testing of Multithreaded Programs and Its Application to Testing Security Protocols*. In Proceedings of the Department of Computer Science University of Illinois at UrbanaChampaign, USA.

[23]. S. Khurshid, C. S. Pasareanu, and W. Visser. 2003, *Generalized Symbolic Execution for Model Checking and Testing*. In Proceedings of the TACAS, Warsaw, Poland.

[24]. S. Sheng, B. Magnien, P. Kumaraguru, A. Acquisti, L. F. Cranor, J. Hong, and E. Nunge. 2007, *Anti-phishing phil: The design and evaluation of a game that teaches people not to fall for phish*. In Proceedings of the 3rd Symposium on Usable Privacy and Security, pp. 88-99.

[25]. P. Kumaraguru, Y. Rhee, S. Sheng, S. Hasan, A. Acquisti, L. F. Cranor, and J. Hong. 2007, *Getting users to pay attention to anti-phishing education: Evaluation of retention and transfer*. In Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit, pp. 70-81.

[26]. S. Garera, N. Provos, M. Chew, and A. D. Rubin. 2007, *A framework for detection and measurement of phishing attacks*. In Proceedings of the 2007 ACM Workshop on Recurring Malcode, pp. 1-8.

[27]. M. Sharifi, and S. H. Siadati. 2008, *A phishing sites blacklist generator*. In Proceedings of the Computer Systems and Applications. AICCSA. IEEE/ACS International Conference, pp. 840-843.

[28]. C. Karlof, U. Shankar, J. Tygar, and D. Wagner. 2007, *Dynamic pharming attacks and locked same-origin policies for web browsers*. In Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 58-71.

[29]. Botnet, http://192.83.193.18/dorm/know-botnet.html

[30]. D.M. Taverira, O.C.M. Duarte. 2008, *A monitor Tool for Anti-spam Mechanisams and Spammers Behavior*. In Proceedings of the Network Operations and Management Symposium Workshops, pp. 101–108.

[31]. SpamAssassin, http://spamassassin.org

[32]. PJ. Sandford, JM. Sandford, and DJ. Parish. 2006, *Analysis of SMTP Connection Characteristics for Detecting Spam Relays*. In Proceedings of the ICCGI '06, pp. 68–68.

[33]. 張倜鈞。兩階層式垃圾郵件過濾機制之研究，私立銘傳大學資訊傳播工程所(2006)。

[34]. ORBD, http://www.coolacid.net/the-news/99-orbdorg-marks-all-as-spam

[35]. I. Biju, J.J. Wendy, and H.S. Jofry. 2009, *Improved Bayesian Anti-Spam Filter – Implementation and Analysis onIndependent Spam Corpuses*. In Proceedings of the ICCET '08, pp. 326-330.

[36]. 資安之眼。全球企業被駭去年平均每家損失200萬美元(2010)。http://www.itis.tw/node/3641

[37]. R.K. Singh, and T. Ramanujam, 2009, *Intrusion Detection System Using Advanced Honeypots*. In Proceedings of the Internatilnal Journal of Computer Science and Information Security - Volume 2, no 1.

[38]. K. Shishir, and P. Durgesh. 2009, *Detection and Prevention of New and Unknown Malware using Honeypots*. In Proceedings of the International Journal on Computer Science and Engineering - Volume 1, no. 2, pp. 56-61.

[39]. M. Christodorescu, and S. Jha. 2004, *Testing malware detectors*. In Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis.

[40]. 張智翔。中央研究院計算中心通訊電子報：淺談網路應用程式安全(二)。http://newsletter.ascc.sinica.edu.tw/news/readnews.php?nid=1294

[41]. Min Gyung Kang, Pongsin Poosankam, and Heng Yin. “Renovo: A Hidden Code Extractor for Packed Executables,” in *5th ACM Workshop on Recurring Malcode (WORM),* October 2007.

[42]. AndroidLib, http://www.androidlib.com/appstatsfreepaid.aspx

五、　計畫架構

異質多網安全檢測平台可支援多種異質網路(例如: WiMAX、3.5G、WiFi無線網路及有線網路)、支援多種行動設備 (例如: 筆記型電腦、迷你筆電、智慧型手機)以及支援多種作業系統 (例如:Windows、Linux)，如圖6所示。

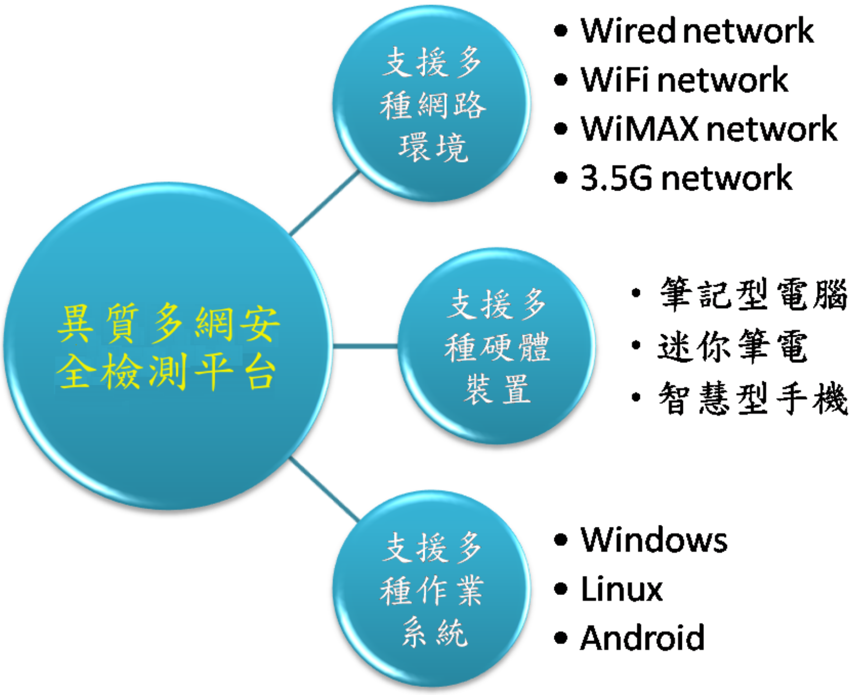


圖 6. 異質多網安全檢測平台支援功能

在2011年，我們開發了以下全新的檢測工具：

- 系統安全檢測類別
  - 線上Windows與Linux系統組態安全檢測系統
  - 作業系統DNS快取毒害監測與防護工具
  - Agent-based合作式滲透測試系統
  - 行動核心網路拓樸探索工具
- 軟體安全檢測類別
  - 線上即時軟體行為分析檢測工具
- 人員安全意識安全檢測類別
  - 社群網路可疑連結隱藏內容發掘系統
  - 行動平台漏洞模擬系統

本計畫將延續過去兩年之成果，持續建置一個異質多網安全檢測平台，提供更為完整、更為全面且可適用於多種異質網路和行動設備之安全檢測工具。另外，我們也將針對不同單位(例如：中華電信、友訊科技等)的需求開發適當且客製化的檢測工具。此外，我們也積極地與其他單位聯繫以及交流，來開發出更符合他們需求的安全檢測工具。

我們預期開發之工具未來可能的合作對象以及承接者如下所述。

線上Windows與Linux系統組態安全檢測系統的檢測項目與行政院研考會規劃的政府機關安全檢測有高度的匹配性，未來此工具將可與中科院以及國家資通安全會報技服中心合作共同開發，以幫助政府部門進行系統組態安全檢測。

作業系統DNS快取毒害監測與防護工具則是藉由同時查詢多台的網域名稱解析器，並透過驗證機制選出一個可信任網路位置的集合，來保護使用者不會受到網域名稱系統快取毒害的威脅。且實作在客戶端，不用修改到任何的域名解析器以及認證伺服器。我們已與中華電信研究所洽談，未來將與該單位進行合作。

線上即時軟體行為分析檢測工具可以提供給電信業者的應用軟體商城(如:中華電信)進行軟體檢測，遏止其上之潛在的惡意軟體的流竄問題。社群網路可疑連結隱藏內容發掘系統可主動提醒使用者網頁連結下之隱藏內容，可提供電信業者(如:中華電信)充實智慧型手機上應用程式之安全性來提升使用者使用意願。

行動平台漏洞模擬系統則是利用多元的系統破解手法找出系統中不同的漏洞，提供工程人員做更全面性的防禦，此系統未來合作對象包括行政院研考會以及國家資通安全會報技術服務中心。此系統將可提供給HIT(Hacks in Taiwan)、資訊工業策進會(金盾獎)做更進一步的精進探討，讓更多的白帽駭客加入系統漏洞的探討。Agent-based合作式滲透測試系統提供一般使用者一個輕便操作系統安全掃描工具，透過瀏覽器連上檢測網即可測知自身電腦暴露在哪些威脅狀況或是自身安全性設定問題，此工具未來合作對象包括行政院研考會、國家資通安全會報技服中心等相關機構。

六、 研究方法

以下我們將根據開發之工具分別介紹其研究方法。

■ **線上即時軟體行為分析檢測工具**

現今惡意程式為了達成其目的，皆試圖延長自身在目標電腦中存活的時間。因此，規避防毒軟體偵測與隱藏自身存在是惡意程式不可避免的行為。但為了達成上述的目標，惡意程式必須對系統中的重要資料進行修改，以開機時自動執行此一常見的行為為例，惡意程式便必須修改註冊機碼中的系統啟動項目，或是在系統ini檔案中加入自身的路徑。因此，若能分析出檢測目標程式在執行後對系統中重要物件的修改行為，便能大幅度提升資安人員在判定時的效率。

本子項目利用Dynamic Information Flow Tracking, DIFT，檢測檢測目標程式對系統中物件的修改行為。整體系統架構為，利用一可追蹤暫存器、記憶體、硬碟磁區以及網路卡中資訊流動的全系統IA-32模擬器，檢測一分鐘後目標程式對系統中重要物件的修改。請注意我們並無法檢測Trigger-based的程式行為，因為若目標程式的行為需要達成某條件才會發生，要自動觸發該行為屬於可被證明為無解的Undecidable的問題。本系統檢測功能如下：

◆ 檔案系統變動監測

當檢測目標儲存至虛擬機器的虛擬硬碟後，便成為本系統追蹤資訊流動的起始點，在運行一分鐘後，所有硬碟中有被該資訊流作用的磁區，將被我們抽取出來，透過解析 NTFS 檔案系統的 metadata，我們可以從這些磁區編號反推回檔案名稱，而得知該檢測目標對系統中那些檔案進行了修改。此作法與其他研究透過監測系統 IO 或系統呼叫以得知修改檔案項目相比，由於後者監測的是系統運行過程中所有的修改檔案項目，而又因作業系統本身及時常對檔案系統有寫入行為，因此有非常多的 False positives，亦即將會有很多並非由於檢測目標所導致的檔案系統修改行為被監測出，而我們的做法由於使用了資訊流動的概念，將正確抓出所有確實由檢測目標所導致的檔案系統修改行為。

◆ 登錄檔變動監測

與前項檔案系統修改檢測功能類似，此分析一樣會透過分析被該資訊流作用磁區的方式，找出登錄檔中被修改的機碼與鍵值。然而此分析目標較困難的是，如何由磁區位址反推回機碼鍵值的步驟。首先，由於登錄檔在硬碟中也未必以連續的方式存放，若僅有磁區編號，要反推為檔案中的偏移位址，也必須先將該檔案在硬碟中所佔據的磁區編號一一取出。此外，Windows 的登錄機碼的儲存方式有一定格式，稱為 Hive File。在 Hive File 中，所有的機碼與鍵值是以二進位的方式儲存，且必須依照特殊格式才能解析，所以我們必須依照 Hive File 格式自行撰寫從鍵值逆推回機碼的演算法，以找出鍵值的所在位址。

在登錄機碼的部分，由於 Windows 的登錄機碼是以特殊的二進位檔的格式，儲存在硬碟中的數個 Hive 檔案中，因此我們是透過觀察這幾個檔案中是否有被寫入，來找出程式所修改的機碼值。這些 Hive 檔包括：

C:\WINDOWS\system32\config\SAM
C:\WINDOWS\system32\config\SECURITY
C:\WINDOWS\system32\config\SOFTWARE

```
C:\WINDOWS\system32\config\SYSTEM
C:\WINDOWS\system32\config\DEFAULT
%UserProfile%\Ntuser.dat
```

由於在找出修改對象的過程必須符合 Hive 檔的格式規定，因此我們將這些檔案中被修改的部分找出來後，在其中搜尋代表機碼值 VKEY 的字樣，若搜尋到則代表此為一真正被分析程式所修改的機碼值。

◆ 程序可執行碼插入(Code Injection) 監測

許多惡意程式為了避免被防毒軟體偵測其行為，往往會將程式碼插入至其他程序後，在該程序的位址空間內執行。透過本系統偵測資訊流動的功能，可檢測出是否有程序間的資訊流動行為，然而僅透過以上作法由於程序間原本就會有訊息交換的行為，因此並無法斷定該資訊流動為程式碼的插入，故我們還需檢查該資訊流在進入其他程序後，是否進一步流動被 CPU 所執行。在判斷程序間的資訊流動方面，我們是透過 CR3 暫存器的變動，以判斷目前正被執行的程序，並在虛擬機器的 CPU 從記憶體中載入程式碼時，判斷該次讀取的記憶體位址是否之前已受到過資訊流動的汙染。此外，僅讀取 CR3 暫存器的值隊分析人員來說並不具備較好的判斷效果，因此本系統還將透過 Hooking 的方式，在任何程序剛建立時建立其程序名稱與 CR3 暫存器的值的對應，以利在產生分析結果時可查閱並顯示被插入可執行碼的程序名稱。

在 Windows 中提供了一項功能，能夠讓一個程式將程式碼插入至其他執行中的程式記憶體中，在該程式的位址空間內執行。透過本工具偵測資訊流動的功能，我們可檢測出是否有程式之間的資訊流動行為，然而由於多個程式之間原本就會有訊息交換的行為，因此僅透過以上作法並無法斷定該資訊流動為程式碼的插入，因此我們還需檢查該資訊流在進入其他程式後，是否進一步流動被 CPU 所執行。在判斷這方面的資訊流動時，我們是透過 CR3 暫存器的變動，以判斷目前正在被執行的程式，並且在虛擬機器的 CPU 從記憶體中載入程式碼時，判斷該次讀取的記憶體位址是否之前已受到過資訊流動的變動。此外，僅讀取 CR3 暫存器對於分析人員來說並不具備較好的判斷效果，因此本工具還透過 Hooking 的方式，在任何程式剛建立時建立其程式名稱與 CR3 暫存器的值的對應，以利在產生分析結果時可查閱並顯示被插入可執行碼的程式名稱。

◆ 作業系統核心可執行碼插入監測

先進的惡意程式多已經使用核心層級的 Rootkit 技術以取得系統最高權限，如此才能躲避同樣具有核心層級權限的防毒軟體的偵測。我們的系統透過觀察資訊流動是否進入核心層級的方式，也可監測此種行為。在 IA-32 架構中，CPU 的執行權限分為 Ring0~3 四種等級，其中 Ring0 為最高權限，而平常的應用程式僅具有 Ring3 等級。在系統運行的過程中，我們將檢查是否 CPU 在執行指令時，是否讀取了被資訊流動污染的記憶體位址，且 CPU 的運行權限處於 Ring 0。

最近許多惡意程式已經開始使用核心層級的 Rootkit 技術以取得系統最高權限，如此才能躲避同樣具有核心層級權限的防毒軟體的偵測。本工具透過觀察資訊流動是否進入核心層級的方式，也可監測此種行為。在 IA-32 架構中，CPU 的執行權限分為 Ring0~3 四種等級，其中 Ring0 為最高權限，而平常的應用程式僅具有 Ring3 等級。在系統運行的過

程中，我們將檢查是否 CPU 在執行指令時，是否曾經讀取了被資訊流動污染的記憶體位址，且 CPU 的運行權限是否處於 Ring 0 層級。

◆ 網路封包監測

透過檢查資訊流動是否流動至網路卡的 TX Buffer，本系統可準確抓出該檢測目標傳送至網路的所有封包。由於現今惡意程式往往會偷取系統資訊，甚至連結外部網站以更新自身，因此側錄這些網路封包是分析工具非常重要的一環。然而，以往的分析工具僅僅是側錄系統運行過程中所有對外的網路傳輸，與第一項檔案系統修改分析類似，在正常的作業系統中，原本就會自行發生許多的網路傳輸行為，例如 Windows Update 的更新檢查、網路芳鄰 Netbios 的查訪、ARP 的更新等等，而這些封包發生的頻率是非常頻繁，以往的分析工具便因此受到大量的干擾或產生 False positives。而我們的系統由於透過追蹤資訊流動的方式，所側錄到的封包皆是確實由一開始的檢測目標，在執行後所產生的，因此具有非常高的準確性與可讀性。

本工具透過檢查資訊流動是否流動至網路卡的 TX Buffer，可準確側錄出所分析程式所傳送至網路的所有封包。由於許多軟體都會連上網路後對自身進行更新，或是接受來自網路的指令，因此側錄這些網路封包是分析工具非常重要的一環。然而，以往的分析工具僅僅是側錄整個系統在運行過程中所有對外的網路傳輸，與第一項檔案系統修改分析類似，在正常的作業系統中，原本就會有許多系統本身的網路傳輸行為，例如 Windows Update 的更新檢查、網路芳鄰 Netbios 的查訪、ARP 的更新等等，而這些封包發生的頻率是非常頻繁，因此以往的分析工具就會有大量的干擾或產生 False positives。而我們的系統由於透過追蹤資訊流動的方式，所側錄到的封包皆是確實由一開始的檢測目標，在執行後所產生的，因此具有非常高的準確性與可讀性。

此外，為了分析人員的便利性，我們將把讀取到的封包依照 TCP/IP 的分層架構做解析，以類似 Wireshark 封包側錄程式的顯示方式列出，如此一來，分析人員可快速的判讀該方包的目的 IP 位址以及所使用的 Protocol。甚至將這些 IP 提報給網路管理員進行黑名單的過濾。

■ **線上 Windows 與 Linux 系統組態安全檢測系統**

本研究的目的在於提供一個線上安全服務用以檢測異質多網環境中終端系統的架構設定安全性。由於終端裝置系統千奇百怪，其上的運行應用程式亦是數以千萬計，並仍不斷在新增中。顯然我們不能單靠一己之力來分析所有可能存在的各種系統組態設定。因此，我們希望能集合眾使用者的力量來匯聚對於各式組態設定的安全資訊並基於這些資訊來對終端使用者提供其系統組態的安全性檢測服務。利用此組態安全檢測工具，異質網路中各式系統均可適用之。

誠如前述，在異質多網環境中充斥著各式各樣的終端裝置系統。而在每個系統上又可能安裝著無數各式的軟體與應用程式。再者使用者可能會對各自的系統或應用程式之設定進行各式各樣的客制化。也因此造成了在異質多網環境下的終端系統之架構設定（configuration）千變萬化的可能。

以系統安全的角度而言，系統的架構設定決定的一個系統先天的安全性。這包括了該系統上是否安裝了具有安全疑慮的軟體元件、惡意程式。再者該系統的各個元件是否有可

能會相衝，導致原設計該有的安全防護功能並沒有如預期的在運作。更重要的是往往一個系統會發生安全問題，背後的原因是由於使用者不當的設定與操作，比如為了方便，對於無線網路、或網路芳鄰不設密碼保護，或同樣為了方便，直接以 root 或 administrator 權限來運行系統等。

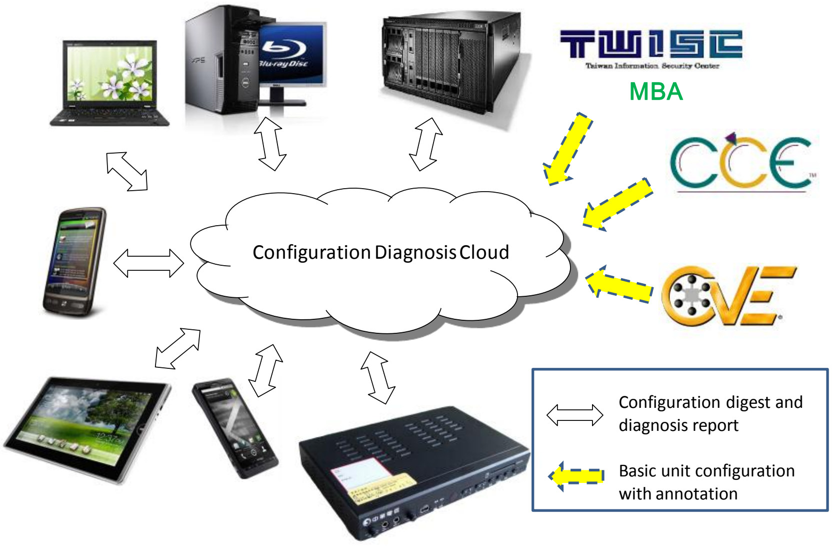

TWISC
Taiwan Information Security Center
MBA
CCE
CVE
Configuration Diagnosis Cloud
Configuration digest and diagnosis report
Basic unit configuration with annotation

圖 7. 線上系統組態安全檢測系統架構

線上系統組態安全檢測系統如圖 7 所示，透過完全對外開放的 API，各個終端裝置提供其各自的架構設定資訊（包括了所安裝的系統元件、應用程式的清單與雜湊值，以及各個元件與程式的設定資料）。所提供的資訊無須包括任何可以用來識別個人資訊的訊息。再者由於 API 是完全公開，使用者可以用各自開發的程式來讀取系統架構設定，也因此無須擔心使用本系統需要於終端系統安裝客端程式（agent），有可能會對系統穩定性造成影響，或是擔心客端程式（agent）可能會內含後門程式等問題。

當檢測完成後，檢測報告亦是透過完全開放的 API 回傳給使用者。使用者可以根據該報告判斷其系統在架構設定上是否具有任何安全疑慮，必要時可根據報告內容採取相應的措施（比如說反安裝有問題的元件、調整相關設定等）。本線上系統組態安全檢測系統亦同時提供介面來匯入對於已知系統架構組態之相關的安全資訊。比如說 CVE 資料庫提供了已知軟體的相關安全弱點資訊、用以確保各系統元件安全性所建議的設定列表 CCE List、以及 TWISC 的惡意軟體分析報告等。

線上系統組態安全檢測系統內部的流程如圖 8 所示。其中兩個輸入介面分別為 X1. Basic unit configuration with annotation。這是供 CVE、CCE、TWISC MBA 等資料庫匯入已知系統元件與架構弱點之資訊。另一輸入介面為 X2，此輸入介面為一般使用者欲對其系統進行檢測時，所必須提供的設定資訊（configuration digest）所走的路徑。一般使用者亦可同時提供他對於該 configuration 的註解（annotation），比如說「本系統中的 web browser 似乎運作不太正常」。使用者提供的註解資訊可以整合進檢測雲的資料庫，以協助對類似系統架構的其他系統進行檢測。

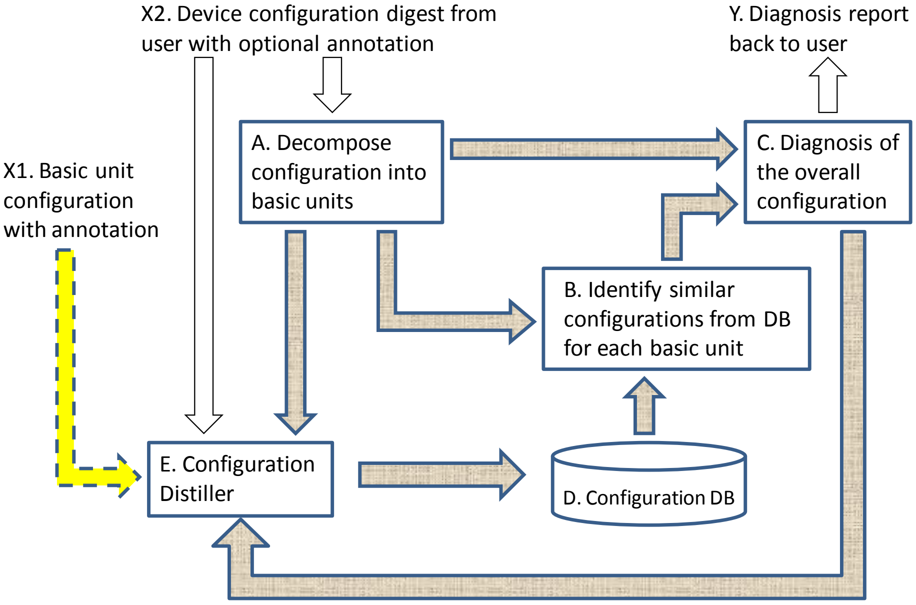

X2. Device configuration digest from user with optional annotation
Y. Diagnosis report back to user
X1. Basic unit configuration with annotation
A. Decompose configuration into basic units
C. Diagnosis of the overall configuration
B. Identify similar configurations from DB for each basic unit
E. Configuration Distiller
D. Configuration DB

圖 8. 線上系統組態安全檢測系統流程

在系統流程中，一般使用者所提供的系統架構資訊 X2 進入系統後的第一步驟是 A，在此步驟使用者的架構資訊以及註解資訊將被轉成系統內部的標準資料結構，並分解為基本組態單元。

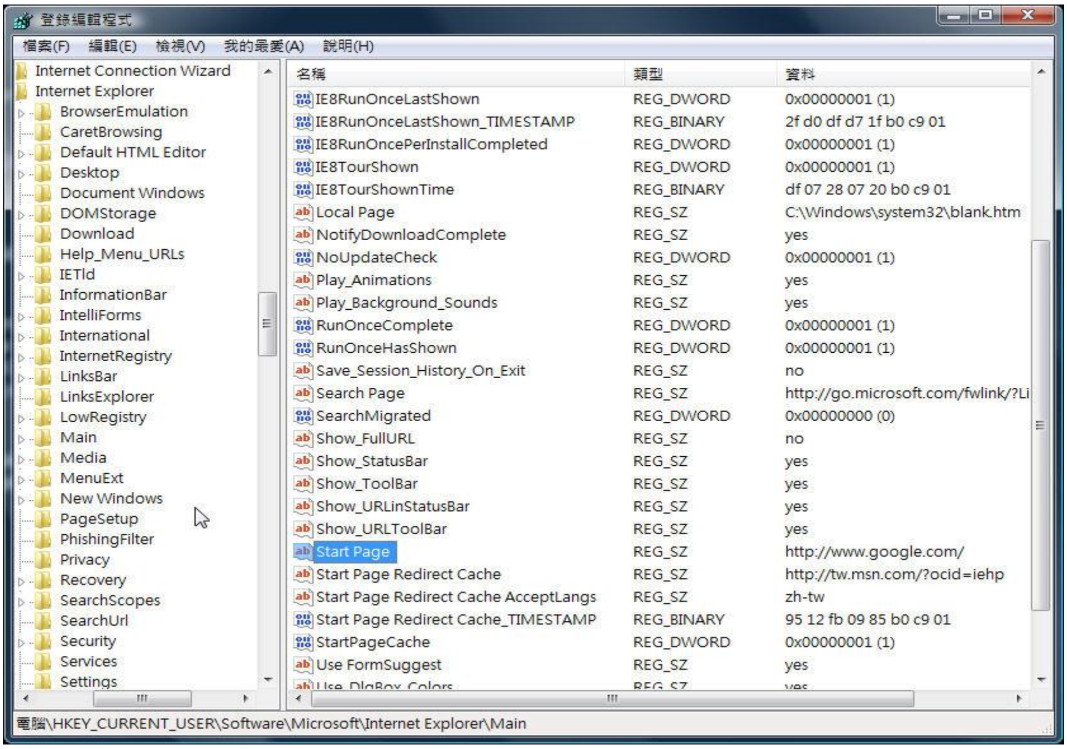
解構為基本單元後的架構資訊會被轉送到步驟 B。在步驟 B 中，各個基本單元的架構資訊將分別與資料庫 D 中已知的基本單元架構資訊進行比對。承襲上段的例子，B 步驟可以從資料庫 D 中擷取對於該版本 Linux 核心的相關安全資訊(比如說該版本核心的記憶體管理有潛在漏洞會造成使用者權限躍升）等。

步驟 C 會根據從步驟 B 所獲取的基本單元之資訊並在考量各基本單元在整體系統中之相互交互關係(由步驟 A 處獲取）來決定對於使用者所欲檢測知系統架構組態提供一個整體的檢測報告。最後將會把檢測報告回傳給使用者（步驟 Y)。

資料庫 D 內的資訊會由步驟 E 中的 Configuration Distiller 持續地擴增與更新。更新資訊的主要來源之一是由專業單位提供(X1)。另外一個渠道則是當一般使用者所提供的檢測資訊(X2)若含有註解資訊（annotation）時，亦可將其併入資料庫。

◆ 系統架構設定通用表示法 (Generic Representation of Configuration)

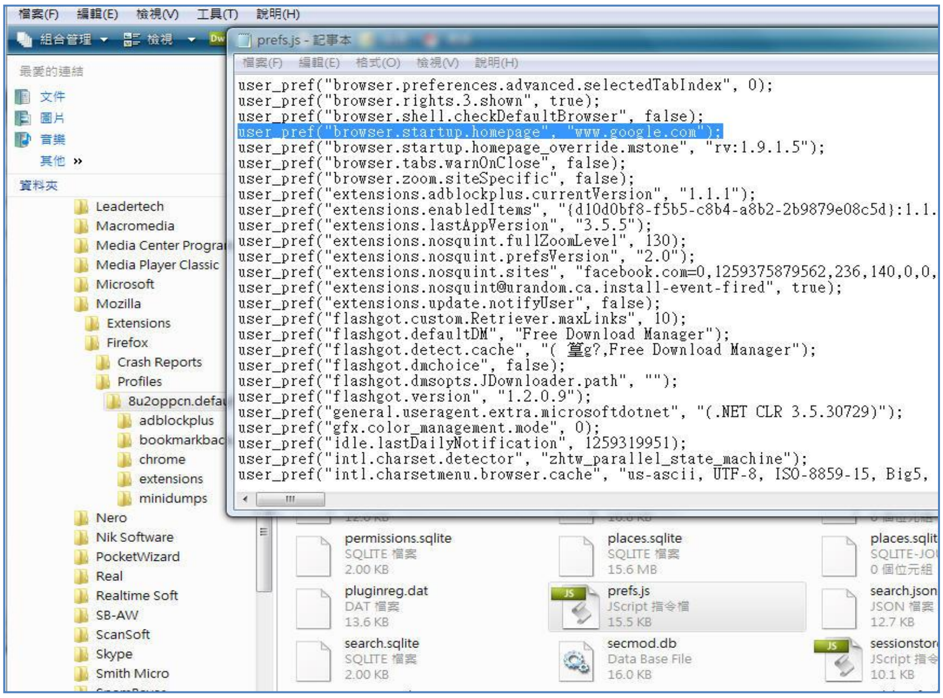
在圖 9 與圖 10 中我們可以分別看到 Internet Explorer 8 跟 Firefox 3.5 兩大主流瀏覽器的架構設定。很明顯的，即便同樣是瀏覽器，兩者的設定資料的呈現方式有著非常顯著的差異性。另一方面，我們亦可發現雖然呈現方式不同，但在概念上兩個瀏覽器的各項架構設定之背後的意涵確有相當程度的相似性。比如說兩著皆可設定瀏覽器預設的首頁。在 IE8 中是以"Start Page"為鍵值的登錄記錄來設定。在 Firefox 3.5 中則是以 browser.startup.hopage 為鍵值儲存於 prefs.js 此一設定檔內。



名稱	類型	資料
IE8RunOnceLastShown	REG_DWORD	0x00000001 (1)
IE8RunOnceLastShown_TIMESTAMP	REG_BINARY	2f d0 df d7 1f b0 c9 01
IE8RunOncePerInstallCompleted	REG_DWORD	0x00000001 (1)
IE8TourShown	REG_DWORD	0x00000001 (1)
IE8TourShownTime	REG_BINARY	df 07 28 07 20 b0 c9 01
Local Page	REG_SZ	C:\Windows\system32\blank.htm
NotifyDownloadComplete	REG_SZ	yes
NoUpdateCheck	REG_DWORD	0x00000001 (1)
Play_Animations	REG_SZ	yes
Play_Background_Sounds	REG_SZ	yes
RunOnceComplete	REG_DWORD	0x00000001 (1)
RunOnceHasShown	REG_DWORD	0x00000001 (1)
Save_Session_History_On_Exit	REG_SZ	no
Search Page	REG_SZ	http://go.microsoft.com/fwlink/?Li
SearchMigrated	REG_DWORD	0x00000000 (0)
Show_FullURL	REG_SZ	no
Show_StatusBar	REG_SZ	yes
Show_ToolBar	REG_SZ	yes
Show_URLinStatusBar	REG_SZ	yes
Show_URLToolBar	REG_SZ	yes
Start Page	REG_SZ	http://www.google.com/
Start Page Redirect Cache	REG_SZ	http://tw.msn.com/?ocid=iehp
Start Page Redirect Cache AcceptLangs	REG_SZ	zh-tw
Start Page Redirect Cache_TIMESTAMP	REG_BINARY	95 12 fb 09 85 b0 c9 01
StartPageCache	REG_DWORD	0x00000001 (1)
Use FormSuggest	REG_SZ	yes
Use DlgBox Colors	REG_SZ	yes

電腦\HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main

圖 9. Internet Explorer 8 開始頁面設定



```
user_pref("browser.preferences.advanced.selectedTabIndex", 0);
user_pref("browser.rights.3.shown", true);
user_pref("browser.shell.checkDefaultBrowser", false);
user_pref("browser.startup.homepage", "www.google.com");
user_pref("browser.startup.homepage_override.mstone", "rv:1.9.1.5");
user_pref("browser.tabs.warnOnClose", false);
user_pref("browser.zoom.siteSpecific", false);
user_pref("extensions.adblockplus.currentVersion", "1.1.1");
user_pref("extensions.enabledItems", "{d10d0bf8-f5b5-c8b4-a8b2-2b9879e08c5d}:1.1.
user_pref("extensions.lastAppVersion", "3.5.5");
user_pref("extensions.nosquint.fullZoomLevel", 130);
user_pref("extensions.nosquint.prefsVersion", "2.0");
user_pref("extensions.nosquint.sites", "facebook.com=0,1259375879562,236,140,0,0,
user_pref("extensions.nosquint@urandom.ca.install-event-fired", true);
user_pref("extensions.update.notifyUser", false);
user_pref("flashgot.custom.Retriever.maxLinks", 10);
user_pref("flashgot.defaultDM", "Free Download Manager");
user_pref("flashgot.detect.cache", "( 簞g?,Free Download Manager");
user_pref("flashgot.dmchoice", false);
user_pref("flashgot.dmsopts.JDownloader.path", "");
user_pref("flashgot.version", "1.2.0.9");
user_pref("general.useragent.extra.microsoftdotnet", "(.NET CLR 3.5.30729)");
user_pref("gfx.color_management.mode", 0);
user_pref("idle.lastDailyNotification", 1259319951);
user_pref("intl.charset.detector", "zhtw_parallel_state_machine");
user_pref("intl.charsetmenu.browser.cache", "us-ascii, UTF-8, ISO-8859-15, Big5,
```

圖 10. Firefox 3.5 開始頁面設定

有鑑於上述的情況，我們體認到需要定義出一個通用的架構設定描述方式。如此方能進行架構設定間的比對以及根據已知的架構設定資料來判斷一個系統的設定是否會具有安全疑慮。

在嘗試定義一套通用架構描述的過程中，我們觀察到一個現象，就是在很多情況下，

一個系統並不是單單僅由其自身所構成。往往在系統內部又可以安裝各式的子系統。如圖 11 所示的是一個運行 Linux 的系統上分別又安裝了兩個「子系統」,OpenOffice 跟 Firefox，而這兩個子系統內部又分別擁有各自的子系統。

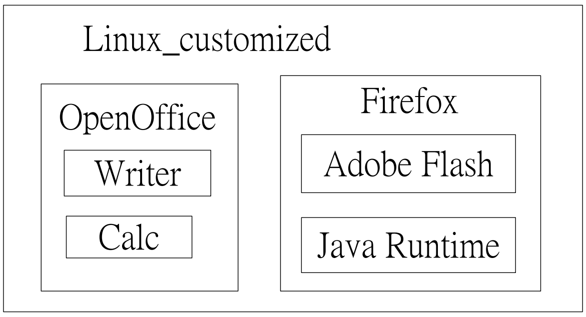
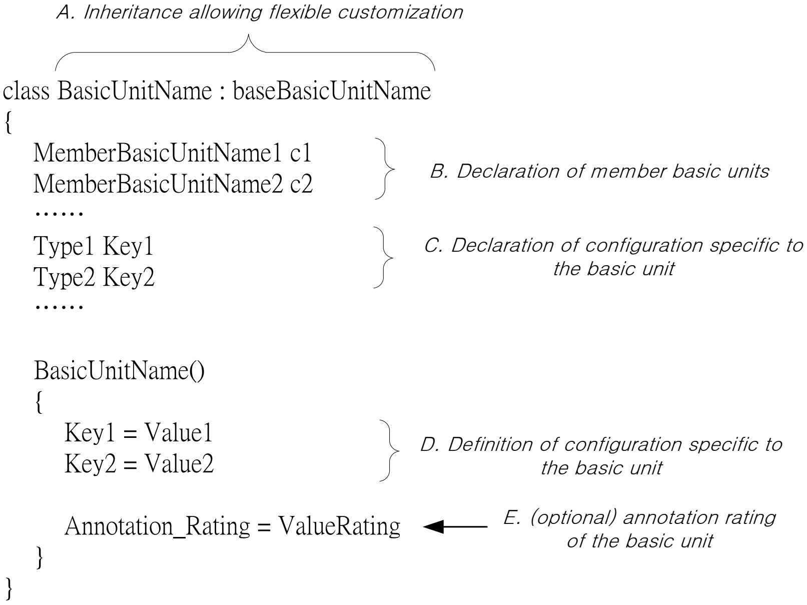


圖 11. Composition of System

我們知道一個系統的安全性並不能單純只看該系統的本體。比如以圖 10 的例子而言。有可能 Linux 本身並無安全漏洞，但或許其內的瀏覽器存在安全漏洞因而造成攻擊者同樣能從遠端侵入該系統。也因此，當我們在描述一個系統的時候不能單純只看該系統本體，也必須同時考慮到期內部子系統構成。


A. Inheritance allowing flexible customization
B. Declaration of member basic units
C. Declaration of configuration specific to the basic unit
D. Definition of configuration specific to the basic unit
E. (optional) annotation rating of the basic unit

```
class BasicUnitName : baseBasicUnitName
{
   MemberBasicUnitName1 c1
   MemberBasicUnitName2 c2
   ……
   Type1 Key1
   Type2 Key2
   ……

   BasicUnitName()
   {
      Key1 = Value1
      Key2 = Value2

      Annotation_Rating = ValueRating
   }
}
```

圖 12. Generic Representation of Configuration

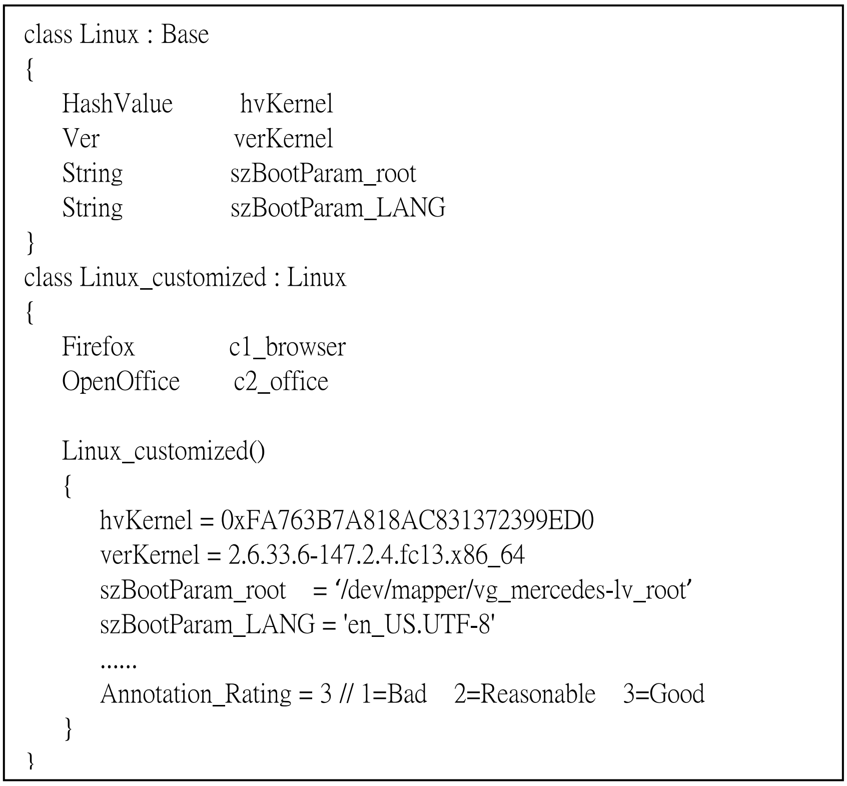
在考量上述等因素後，我們所設計的系統架構通用描述如圖 12 所示。讀者亦可同時參照針對圖 13 中系統所對應的一個實際的通用描述範例。

首先我們對於系統內的每個子系統以及該系統本身皆稱其為一個「基本單元」(Basic Unit)。以上述的範例系統來說，我們分別有 Linux_customized、Firefox、OpenOffice、Writer、Calc、Adobe Flash、Java Runtime 等七個 basic unit。在圖 12 中 A 處所看到的是基本單元可以繼承一個父基本單元，這個的目的是可以用來免除對於共通之架構設定的重複定義。比如說運行 Linux 的系統皆會有系統核心，也因此針對系統核心相關的設定可以將其放置一個父親基本單元中（如圖 13 中之 Linux class）。

緊接著在 B 處我們看到的是對於內部的子系統的宣告。比如對應圖 13 我們可以看到針對範例系統中 Firefox 跟 OpenOffice 所宣告的兩個子系統 c1_browser 與 c2_office。

在 C 處是對於該基本單元所特有的架構設定的宣告。比如在圖 13 中 Linux 基本單元的宣告內容中我們可以看到分別對於核心映象檔的雜湊值 hvKernel、核心版本 verKernel 等設定值的宣告。這些架構設定的值則是在 D 處中所指定。

在 E 處是使用者對於該系統的註解（annotation）。以目前來說，我們考慮最簡單的情況，也就是使用者會給該系統打個評價，數值越高代表系統越安全。以實際應用來說，我們可以區分一般使用者跟受信任的專業使用者(如 TWISC MBA、CVE、CCE)的 annotation，以避免專業知識不足的使用者或是具惡意的使用者故意提供假的 annotation 所可能會對系統整體造成的影響。



```
class Linux : Base
{
    HashValue        hvKernel
    Ver              verKernel
    String           szBootParam_root
    String           szBootParam_LANG
}
class Linux_customized : Linux
{
    Firefox          c1_browser
    OpenOffice       c2_office

    Linux_customized()
    {
        hvKernel = 0xFA763B7A818AC831372399ED0
        verKernel = 2.6.33.6-147.2.4.fc13.x86_64
        szBootParam_root    = ‘/dev/mapper/vg_mercedes-lv_root’
        szBootParam_LANG = 'en_US.UTF-8'
        ......
        Annotation_Rating = 3 // 1=Bad    2=Reasonable    3=Good
    }
}
```

圖 13. Sample generic representation of the configuration for the system

◆ 系統架構組態之安全性診斷

當使用者所上傳的系統架構設定解構成基本單元後(圖 8 步驟 A)，本系統會對每個基本單元分別與資料庫進行比對以計算該單元的安全性評價。比對的原則是從資料庫中找出與該基本單元類似架構的已知單元，並用那些相似單元的註解資訊來分析該單元是否安全。對於尋找「相似」的單元，我們考量三個點。第一點是兩個單元間的通用描述中繼承架構的相似性。比如說若兩個單元皆為 Linux 所衍伸的系統，其相似度就會比一個 Linux 所衍伸的系統單元與一個 Windows 所衍伸的系統單元彼此間的相似度來得高。第二點我們會考量兩個單元內部子單元的構成（圖 8 B 處）的相似度。第三點則是考量兩個單元內部設定的相似度（圖 12 C、D 處）。後面這兩點是出於前段有提到的，一個系統的安全性並不

單純取決於自身而已，而必須同時考量到該系統內部的配置情況（包括了擁有哪些子系統以及相關的設定等）。

透過相似度的比對，對於一個基本單元 Ck 我們會從資料庫得到一群與之類似的已知基本單元 S1, S2, ..., Sn。接下來對於基本單元 Ck 安全性的評價我們預計先採用加權平均作法，也就是把各個相似單元的註解評價(annotation_rating)透過加權(W(.)函數)平均而得之。加權的目的在於降低資料庫中可信度較低的資料所可能造成的影響（比如說由專業經驗不足使用者所提供的註解資料之可信度就相對較低）。

◆ 資料庫內容的新增與更新

本系統一方面是靠使用者的分享資料來構成，這部分在概念上雷同於比對多台 Windows 機器設定檔以對問題機器進行架構設定檢測的作法。我們的創新處在於將此概念推廣到異質多網系統，以及集結分布各地的使用者來齊力達到檢測各類系統設定安全的目的。當一個使用者在上傳欲檢測的架構組態資訊時，他可以同時上傳相對應的註解資訊。以初期來說，此項大部分必須仰賴使用者自發性的資訊分享。長期來說，我們可以嘗試整合各類的系統內建錯誤報告（bug report）或使用者經驗回饋 （user experience feedback）等機制來擴充由此管道所能得到的新資料。

在另一方面，我們亦可由專業的機構來提供對於各系統相關安全性的資訊。比如說 TWISC MBA 提供惡意軟體的檢測，如果確定了某隻應用程式為惡意軟體，那麼這個資訊就可以直接整合進我們的資料庫。再者諸如 CVE、CCE 等均是我們可以利用的專業安全資訊來源。

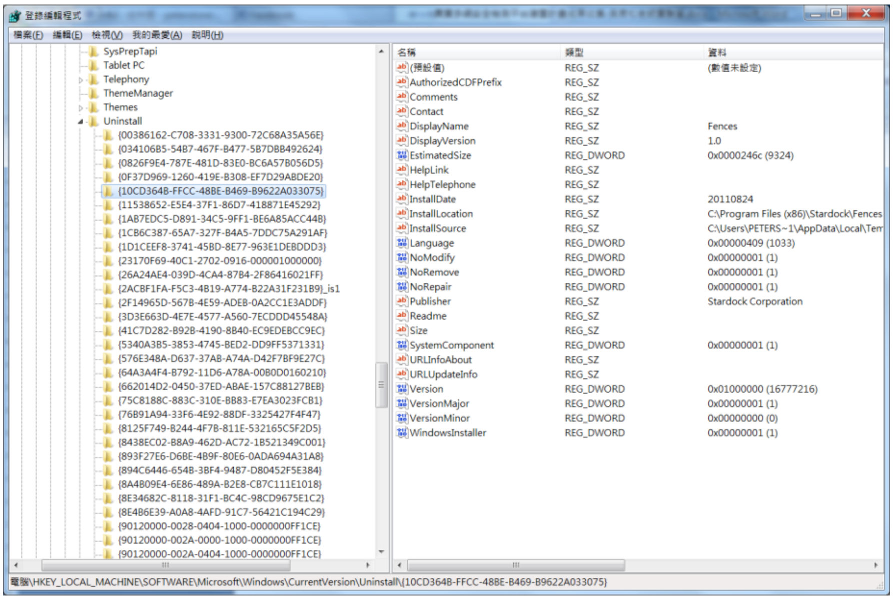
關於系統組態安全的研究，可分兩部份來說明：系統組態的內容，以及何謂安全的組態。另外關於每個系統的組態，我們用 XML 格式的檔案來描述，詳細內容亦會於以下篇幅內說明。

● 系統組態

不論是個人電腦或是行動裝置，每個系統都可以看作是由很多元件組合而成：作業系統上安裝了哪些軟體、各個軟體各自使用到哪些套件等等。在此計畫中，我們以兩個作業系統為研究對：Windows、Linux。其中又聚焦於 Windows 作業系統。因為 Windows 系統有特有的登錄檔管理機制，所以在 Windows 上我們以登錄檔為主要資訊來源，並對其進行研究分析。

■ Windows

首先我們蒐集系統上已安裝的軟體資訊，就像是 Windows 控制台裡的「解除安裝/變更程式」列表裡列出的目前系統中存在的軟體。我們紀錄軟體資訊及版本號等資訊。相關的資訊在登錄檔上是有跡可循的，「HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall」這個 Registry Key 底下提供了我們可參考的資訊，如圖 14。



```
SysPrepTapi
Tablet PC
Telephony
ThemeManager
Themes
Uninstall
    {00386162-C708-3331-9300-72C68A35A56E}
    {034106B5-54B7-467F-B477-5B7DBB492624}
    {0826F9E4-787E-481D-83E0-BC6A57B056D5}
    {0F37D969-1260-419E-B308-EF7D29ABDE20}
    {10CD364B-FFCC-48BE-B469-B9622A033075}
    {11538652-E5E4-37F1-86D7-418871E45292}
    {1AB7EDC5-D891-34C5-9FF1-BE6A85ACC44B}
    {1CB6C387-65A7-327F-B4A5-7DDC75A291AF}
    {1D1CEEF8-3741-45BD-8E77-963E1DEBDDD3}
    {23170F69-40C1-2702-0916-000001000000}
    {26A24AE4-039D-4CA4-87B4-2F86416021FF}
    {2ACBF1FA-F5C3-4B19-A774-B22A31F231B9}_is1
    {2F14965D-567B-4E59-ADEB-0A2CC1E3ADDF}
    {3D3E663D-4E7E-4577-A560-7ECDDD45548A}
    {41C7D282-B92B-4190-8B40-EC9EDEBCC9EC}
    {5340A3B5-3853-4745-BED2-DD9FF5371331}
    {576E348A-D637-37AB-A74A-D42F7BF9E27C}
    {64A3A4F4-B792-11D6-A78A-00B0D0160210}
    {662014D2-0450-37ED-ABAE-157C88127BEB}
    {75C8188C-883C-310E-BB83-E7EA3023FCB1}
    {76B91A94-33F6-4E92-88DF-3325427F4F47}
    {8125F749-B244-4F7B-811E-532165C5F2D5}
    {8438EC02-B8A9-462D-AC72-1B521349C001}
    {893F27E6-D6BE-4B9F-80E6-0ADA694A31A8}
    {894C6446-654B-3BF4-9487-D80452F5E384}
    {8A4B09E4-6E86-489A-B2E8-CB7C111E1018}
    {8E34682C-8118-31F1-BC4C-98CD9675E1C2}
    {8E4B6E39-A0A8-4AFD-91C7-56421C194C29}
    {90120000-0028-0404-1000-0000000FF1CE}
    {90120000-002A-0000-1000-0000000FF1CE}
    {90120000-002A-0404-1000-0000000FF1CE}
```

名稱	類型	資料
(預設值)	REG_SZ	(數值未設定)
AuthorizedCDFPrefix	REG_SZ	
Comments	REG_SZ	
Contact	REG_SZ	
DisplayName	REG_SZ	Fences
DisplayVersion	REG_SZ	1.0
EstimatedSize	REG_DWORD	0x0000246c (9324)
HelpLink	REG_SZ	
HelpTelephone	REG_SZ	
InstallDate	REG_SZ	20110824
InstallLocation	REG_SZ	C:\Program Files (x86)\Stardock\Fences
InstallSource	REG_SZ	C:\Users\PETERS~1\AppData\Local\Tem
Language	REG_DWORD	0x00000409 (1033)
NoModify	REG_DWORD	0x00000001 (1)
NoRemove	REG_DWORD	0x00000001 (1)
NoRepair	REG_DWORD	0x00000001 (1)
Publisher	REG_SZ	Stardock Corporation
Readme	REG_SZ	
Size	REG_SZ	
SystemComponent	REG_DWORD	0x00000001 (1)
URLInfoAbout	REG_SZ	
URLUpdateInfo	REG_SZ	
Version	REG_DWORD	0x01000000 (16777216)
VersionMajor	REG_DWORD	0x00000001 (1)
VersionMinor	REG_DWORD	0x00000000 (0)
WindowsInstaller	REG_DWORD	0x00000001 (1)

電腦\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{10CD364B-FFCC-48BE-B469-B9622A033075}

圖 14.　登錄鍵值範例

在以上 registry key 裡包含許多 subkey，每個 subkey 都包含一個軟體或是套件的相關資訊，如名稱、版本號、安裝目錄等等。針對此列表，我們依據各自的軟體/套件名稱做一些分類。因 Windows 上微軟相關的軟體種類繁多，所以基本上區分為兩大類別：與微軟有關以及其他，在 XML 檔案裡分別對應到「Microsoft」及「OtherApplications」此兩節點。

■　Linux

在以 Unix/Linux 為基礎的系統上，主要是透過各系統上所擁有的套件管理程式，去取得我們想要獲取的資訊。舉例來說，像是在 Redhat 系列之系統上，我們可以利用 RPM 這個套件管理系統去取得系統所存在之應用程式、應用程式版本，甚至可知道程式間彼此的相依性等等。而在 Debian 系列之系統上，也可以透過 APT 套件管理系統取得類似的資料。在取得系統應用程式的資料後，同樣的我們會將之轉換為我們所上述所說的 XML 組態描述檔。

●　安全的組態

在判斷組態是否安全的議題上，我們分兩方向進行。一為規則式的檢測，二為統計式的檢測。在規則式的檢測方面，我們依據 XML 組態檔的部份內容分析出一些規範。舉軟體更新為例，一般而言，若系統有安裝 Office 2007，則大致上亦會有 Office 2007 相關的更新套件。我們視此類規範為部份檢測的標準。

在統計式的檢測方面，我們仰賴使用者的群體智慧。若是我們在大部分使用者的組態檔中皆發現有哪個項目或內容，我們就將之定為檢測的新規則。其原理類似於 Data Mining 裡尋找 Frequent Sequence。當然在此部份中，support 值（成為 frequent item 的門檻）就有很多實驗的空間。

■　**作業系統 DNS 快取毒害監測與防護工具**

針對 DNS Cache Poisoning 攻擊，我們提出了一套在使用者端的防禦機制。主要架構

為，在當使用者瀏覽網頁的同時，本系統會主動向多台 DNS Resolver 發出詢問封包，再透過分析各個 Resolver 回傳的結果，對回傳來的 IP 做分類並且評分。評分過了一定的標準後，才會被我們認可為受信任的 IP。系統架構如圖 15 所示。

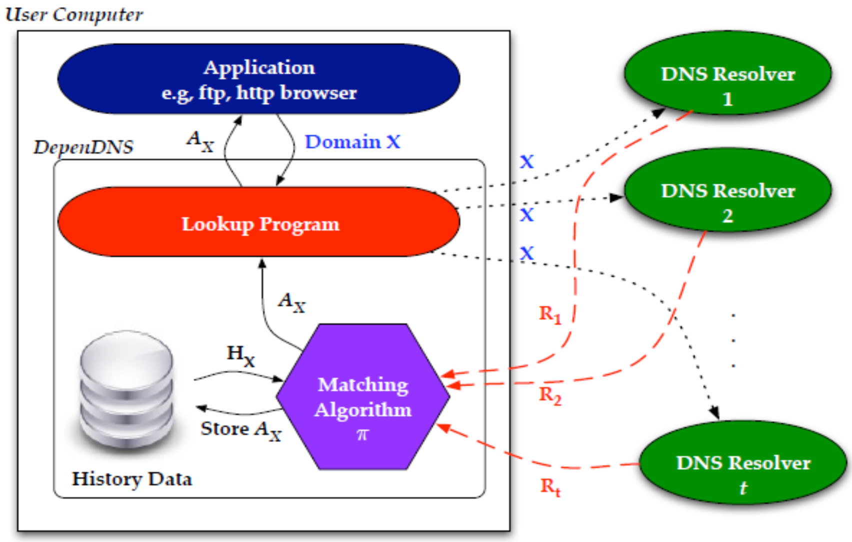


圖 15. 網域快取毒害線上防護系統架構

為了達到跨平台與減少流量的目的，我們將發送詢問封包的機制移至架設於 TWISC 的 DepenDNS Server 上，當使用者透過網頁瀏覽器或智慧型手機應用程式瀏覽網站時，會向 DepenDNS Server 詢問結果，而 DepenDNS 會負責去詢問多台 DNS Resolver，最後將分數及過標準的 IP 回傳給使用者的瀏覽器。

DepenDNS 主要可以分成三大部分

1. 發送 DNS 詢問封包給個選定的 DNS Resolvers·
2. 透過 Algorithm π 來計算第一部分回傳的結果·
3. History Database 來記錄以往的歷史資料·

第一部分，透過選定台灣各大 ISP 業者提供的開放 DNS Resolver，本系統選最後選定六台 DNS Resolver，原因是為了縮短詢問等待的時間，二來也是減少網路流量。而選定的 DNS Resolver 也可以直接修改客製化而不需修改程式。

第二部分，我們所提出一套 Algorithm π 來運算個 IP 的分數。詳細方法說明如下

Algorithm π 主要可分成三個部分: α, β, γ 三種分類，α 佔 60%，β, γ 各佔 20%。

α 所做的分析為將所有從 DNS Resolver 拿到的 IP 做分類比較，差異度超過 20%以上的則拿 0 分，小於 20%的則拿 1。其定義如下：

$$\alpha = \begin{cases} 1, \text{ if } n^i \geq n^{max}(1 - 20\%). \\ 0, \text{ otherwise.} \end{cases}$$

會取 20%的主因是由於，有些網域會有負載平衡的機制，同一個網域名稱會對應到不只一組 IP，如此我們可以容許 20%的差距。

β 為歷史資料的結果，從 DNS Resolver 取得的 IP 如果出現在歷史資料中，β 則為 1，

反之則 0。其定義如下

$$\beta = \begin{cases} 1, \text{ if IP}_i \text{ exists in history data at the same domain name.} \\ 0, \text{ otherwise.} \end{cases}$$

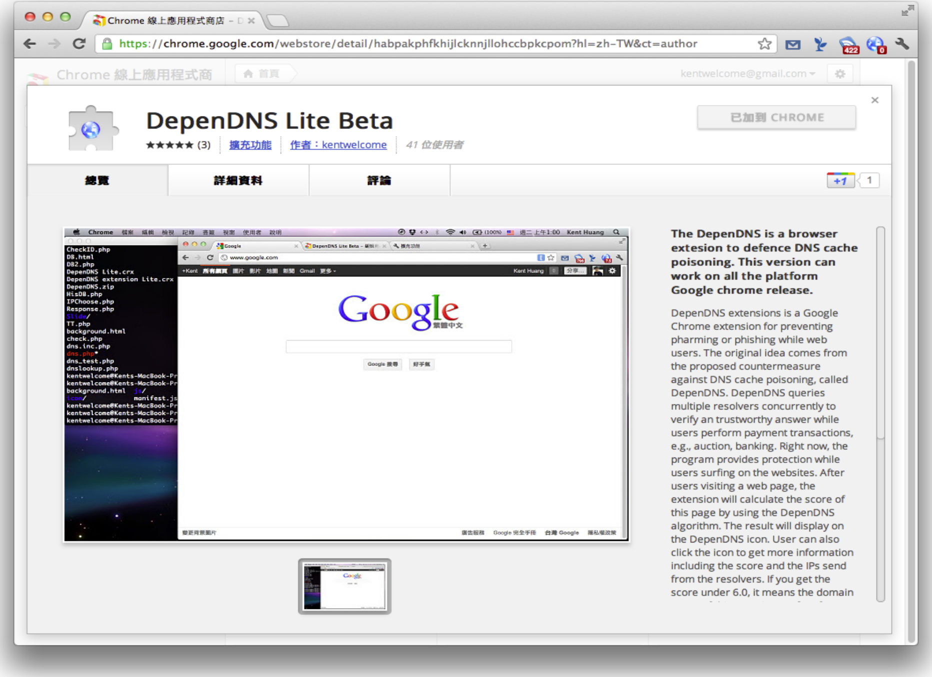
γ 則針對歷史資料的 IP 做分析，若 DNS Resolver 所取得的 ip 雖然不再歷史資料中，可是其 IP Class B 分布的差異不超過 10%，則 γ 為 1，若超過 10%以上則為 0。定義如下

$$\gamma = \begin{cases} 1, \text{ if IP}_i \text{ belongs to } k^{th} \text{ class B and} \\ \quad -10\% \leq c^k_{current} - c^k_{history} \leq 10\% \\ 0, \text{ otherwise.} \end{cases}$$

最後總分的算法，則依照 α, β, γ 以 60%，20%，20%的比重，算出結果。

$G = \alpha * (G\alpha \; - (N - 1) * 10)\% + ½ *(\beta + \gamma)*(G\beta\gamma \; + (N -1) * 10)\%$ 系統會將所得分數超過 60 分的所有 ip 回傳給使用者。本系統使用 PHP 搭配 MySQL 資料庫見至於 Apache web server 的環境上。

由 DependDNS Server 所提供的服務，我們分別針對 Google Chrome 瀏覽器和 Apple iPhone 應用程式，開發可以和 DepenDNS Server 連線的程式。Google Chrome 瀏覽器外掛已經上傳至 Google Web Store 開放全球使用者下載。使用者安裝後，每次瀏覽器再瀏覽網頁時，經由 DepenDNS Server 所運算的分數會顯示於畫面的右上角，而按下 DependDNS 按鈕後會顯示目前瀏覽網站的安全 IP 列表及所得的分數。DepenDNS on iPhone 則為 iOS 原生應用程式，使用者點選此應用程式後，便可以透過此 App 瀏覽網頁，在瀏覽網頁同時，手機會連上 DepenDNS Server 去詢問安全的 IP，最後比對手機上所反查的 ip 和安全名單中的 ip 是否吻合。圖 16 為所實作出之系統運行時之截圖。



The DepenDNS is a browser extesion to defence DNS cache poisoning. This version can work on all the platform Google chrome release.

DepenDNS extensions is a Google Chrome extension for preventing pharming or phishing while web users. The original idea comes from the proposed countermeasure against DNS cache poisoning, called DepenDNS. DepenDNS queries multiple resolvers concurrently to verify an trustworthy answer while users perform payment transactions, e.g., auction, banking. Right now, the program provides protection while users surfing on the websites. After users visiting a web page, the extension will calculate the score of this page by using the DepenDNS algorithm. The result will display on the DepenDNS icon. User can also click the icon to get more information including the score and the IPs send from the resolvers. If you get the score under 6.0, it means the domain

圖 16. DNS 快取毒害線上防護系統截圖

■ **社群網路可疑連結隱藏內容發掘系統**

在此次計畫中，我們的程式以 Google chrome extension 的方式呈現。Google chrome 是 Google 在 2008 年發布的瀏覽器，具穩定、速度、安全及輕便為主要訴求，使用者人數目前約佔 15%，已成為主流瀏覽器之一。Google chrome extension 為瀏覽器的擴充功能，使用者可以依照自己喜好使用，也可以根據需求開發新的擴充功能。由於我們的程式以網頁應用為主，將程式與瀏覽器擴充功能結合，可以讓使用者在使用我們開發的工具時不需要執行額外的程式，增加使用者使用的意願。

最初，我們希望設計能自動幫使用者過濾會散佈廣告訊息到塗鴉牆的連結（「讚」），讓使用者不會因為好奇讓塗鴉牆變成免費的廣告看版；然而，阻擋這些連結並不能滿足使用者的好奇心，甚至可能有使用者為了觀看隱藏內容而寧可讓對方在塗鴉牆發佈訊息。因此，我們改變想法，讓使用者能在不按「讚」的情況下依然能觀看隱藏內容，這樣不但能滿足使用者的好奇心，也能讓使用者防止自己的塗鴉牆成為廣告看版。

在我們的設計中，我們提供讓使用者自行決定是否要顯示隱藏內容或阻擋 Facebook 的功能。透過點擊 Google chrome 右上角的 popup page icon，畫面中會跳出互動式訊息視窗，若使用者點選 Yes，則 background page 會開始進行處理，然後在網頁中顯示被 Facebook 元件隱藏住的隱藏內容；若使用者點選 No，則 background page 會開始檢查所有送出的 HTTP request，如果其中包含 Facebook 相關的操作，此連線請求將會被阻擋。

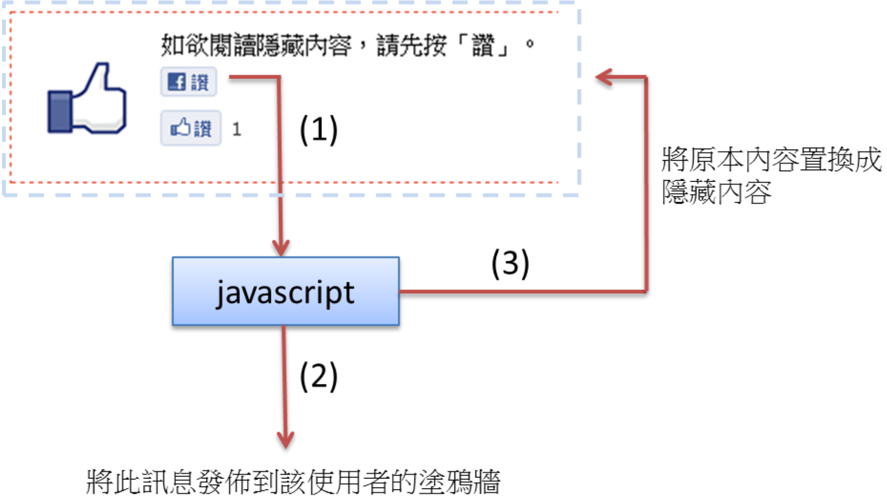


圖 17.社群網路可疑連結隱藏內容發掘系統運作流程

為了在不按「讚」的情況下顯示隱藏內容，必須先了解隱藏內容所使用的機制，如圖 17 所示。根據檢測結果發現，比較常見的隱藏內容是將欲隱藏的文字、圖片或影片放在某段 JavaScirpt 內，當使用者點擊網頁上的「讚」或「分享」之後，此段 JavaScript 便會被執行，而這段 JavaScript 主要會做以下兩件事：(1)、發佈一則訊息至該使用者的塗鴉牆（如果是 facebook）或其他可能有害的動作（ex：跳出廣告、跳出具攻擊性的網頁）。(2)、清空原本隱藏住的地方並顯示隱藏內容。

(2)就理論上來說可以不需要被執行，但從社交工程的角度來看，若(2)沒執行（沒顯示隱藏內容），便等同於是使用者單方面受騙，久而久之就不會再有使用者受騙點擊。因

此「回饋使用者的好奇心」便是隱藏內容機制的精隨，同時也是破綻所在。為了回饋使用者點擊完該顯示的內容，此內容大部分會隱藏在和 JavaScript 相關的程式當中，因此只要能從網頁原始碼中找出使用者想看到的隱藏內容，再仿照 JavaScript 的顯示方式呈現，便能做到讓使用者不需要點擊「讚」或其他連結就能看到網頁中被隱藏的內容。

要找出隱藏在原始碼中的隱藏內容連結，其中一個方法是利用 pattern database，搜集各種網頁所採用的 JavaScript pattern 並存在資料庫中，然後在需要顯示隱藏內容時利用搜集到的 pattern 進行比對，進而找出該網頁所隱藏的內容。因此，pattern 的數量與精確度以及字串搜尋比對的準確度將是這項工作成功與否的關鍵。

- **行動平台漏洞模擬系統**

隨最近 Android、iPhone…等手機平台陸陸續續都已出現了首隻的病毒，可見手機平台的漏洞將是最新的攻擊方式，而且，以往在個人電腦或伺服器方面，因安全性問題遭受實質損失者，大多為企業居多；但隨著近期智慧型手機的流行，加上手機通訊服務是收費平台，對於一般的民眾，這些相關的智慧型手機漏洞造成的損失，甚至可能比電腦安全性問題更為嚴重，因此基於 mobile OS 的安全問題，將是未來我們必須重視的目標。

我們將以 Virtual Machine 為基礎的架構，開發以行動平台的作業系統為主的程式安全訓練平台，讓使用者能在 Android、iOS(iPhone OS)…等行動平台的操作環境上，利用我們設計的行動平台相關類型的題目，例如 iOS 上的 Jailbreak 破解…等，進行實際的訓練。透過如此實際的操作，讓使用者能從中了解其中的原理。本訓練平台不僅提供安全訓練的環境，也會透過提示來引導使用者解題的方向，大大地降低解題的門檻，提升初級使用者的興趣。如果解題遇到困難或瓶頸，亦可透過相關的論壇來與其他使用者交流、切磋、探討解題方向是否有偏差，以及其中學習其他人解決的方法。我們的平台設計盡量滿足進階與初階使用者的需求，將來會開發與提供多樣化的作業系統和環境，讓使用者體驗更多元化解題技術的同時，也不用承擔對系統攻擊的法律追究。

隨著目前 Virtual Machine 相關技術的發展越來越成熟，讓 Virtual Machine 執行的速度更快、效能更好，目前在非 I/O 的操作上甚至可以達到接近原來執行的速度。因此我們將利用 Virtual Machine 的特性來開發程式安全訓練平台，使得能為每位使用者提供各自的單獨的一個作業系統，讓使用者單獨面對個人化的操作環境。因為每個使用者的操作環境是獨立的，在題目的變化性上自由度更高，例如對於作業系統等級上的安全性問題，也能透過這個架構，產生相對應的題目，讓使用者能真實攻擊一個有安全性問題的作業系統，卻不影響其他使用者與訓練平台的運作。

最後，未來這個行動平台模擬系統能夠與一般的 Wargame 服務整合，提供針對 Mobile Platform 相關的程式安全訓練，擴展一般 Wargame 服務的範圍。另外，無論是 Desktop 平台或 Mobile 平台的作業系統常常有安全性的 patch，未來與 Wargame 整合後，也能透過收集這些各版本 patch 之前的有問題作業系統，提供類似作業系統 database 的服務。利用 Virtual Machine 的 snapshot(安全快照備份)功能可以很方便的在每次 patch 前進行備份。當使用者想測試某個版本或某個有問題的作業系統，也可以很方便的從各 snapshot 備份版本中挑選出來，利用 Virtual Machine 進行 boot，提供使用者使用，且因為有 snapshot 的功能，系統可以快速的將作業系統 clone 一份，進行 boot 的動作，確保不會影響 Wargame 中保存

的原生(base)環境，也不會影響其他使用者的環境，達到權限隔離的效果。

- ◆ Virtual Machine 平台的架設

隨著目前 Virtual Machine 相關技術的發展越來越成熟，讓 Virtual Machine 執行的速度更快、效能更好，目前在非 I/O 的操作上甚至可以達到接近原來執行的速度。因此我們將研究如何利用 Virtual Machine 的特性來開發程式安全訓練平台，使得能為每位使用者提供各自的單獨的一個作業系統。讓使用者單獨面對個人化的操作環境。因為每個使用者的操作環境是獨立的，可避免使用者之間的互相干擾，在題目的自由度上變化也更高，例如對於作業系統等級上的安全性問題，也能透過這個架構，產生相對應的題目，讓使用者能真實攻擊一個有安全性問題的作業系統，卻不影響其他使用者與訓練平台的運作。應用此技術另外一個優點則為可快速佈署不同版本的 Android 版本，讓使用者也可同時於多個不同版本的 Android 系統上進行關卡訓練，不會受限於僅能操作特定版本的作業系統版本。

而設計關卡時，必須先考慮到 Android 系統的架構，以便於在開發應用程式時，得以和 Android 介面互動，而 Android 為了降低開發者對於平台的進入門檻，因而在應用程式內與內部系統的環境之間設置了一層應用程式框架(Application framework)的中介介面，讓程式可以更加容易溝通，對於我們在開發程式來說，如需要實作底層的函示庫(Libraries)功能，則可以直接利用 Java 程式來呼叫應用程式框架所提供的應用程式介面(Application Programming Interface, API)即可。在測試的開發環境時，我們是在 Android 手機模擬器上來模擬實際環境，進而開發出適合初階與進階使用者的關卡。

- ◆ ARM 與 x86 平台相容性測試與移植

由於我們所設計的關卡實際上是運作於 x86 的 Virtual Machine 中，程式在 x86 平台中運行的結果會與實際上在 ARM 平台上的 Android 行動裝置上有些許差異，為確保我們在 x86 平台上所設計的關卡，必須將在 Android Emulator 上所撰寫的程式先實際拿到 ARM-Based 的 Android 行動裝置上測試，如可正常運作後，再將該程式移植至我們所建立的 Virtual Machine 中運行，看其結果是否如設計者所預期一般，如有相異之處，則必須再就有問題的地方進行研究，確保使用者在平台上所學到的知識是可以實際上運用於 Android 行動裝置中的。

- ◆ Android 相關漏洞研究與設計

由於要設計與 Android 行動裝置漏洞有關的關卡，一開始我們會先蒐集網路上一些現有的 Android 漏洞並針對該漏洞進行研究，並根據該漏洞的特性來設計相對應的關卡。

在模擬平台關卡的設計上，我們主要會採兩個方式進行透過現有漏洞設計而成的關卡，主要是針對初學的使用者，因在網路上已可找到許多人對於該漏洞所做的分析與講解，對於初學者來說，不需較多對於 Android 系統艱澀的知識與研究，無非是一個較為容易的進入門檻，使得初學者得以透過入門關卡建立信心。而在閱讀這些資料的過程中，使用者也需慢慢去補齊相關的 Android 知識，使得使用者可以逐步開始深入了解 Android 的運作，進而挑戰我們所自行設計的 Android 漏洞。而自行設計的 Android 漏洞關卡都是由模擬平台的開發者所自行設計的，常見的軟體安全問題都會被包裝成一關一關的關卡，供使用者逐一挑戰。

由於平台的另一功用為程式漏洞驗證與測試，因此我們也打算在將來針對這部分開發

出一個介面供使用者自行上傳可疑的程式至模擬平台上，讓其他進階的使用者得以對於這些程式進行挑戰，當找出潛在性的問題時，在回報於該程式的上傳者，讓他得以驗證程式的正確性。

■ **Agent-based 合作式滲透測試系統**

弱點掃描或是滲透測試是常見的系統安全檢測方法。滲透測試以攻擊者的角度，真實地去嘗試滲透攻擊使用者的系統；若是滲透攻擊成功侵入使用者系統，便明確地指出威脅使用者系統安全的對應漏洞。利用滲透測試檢測使用者系統，容易受到網路防火牆設置的影響，無法完整地找出使用者系統中所有的安全漏洞，形成使用者系統潛在的安全隱憂。另一方面，滲透測試透過使用者系統對外連線的網路介面進行滲透攻擊，使用者系統如果具有其他的網路介面，滲透測試將無法對其他的網路介面進行滲透攻擊，也無法找出相對應的系統安全漏洞，同樣成為使用者系統內潛在的安全危機。

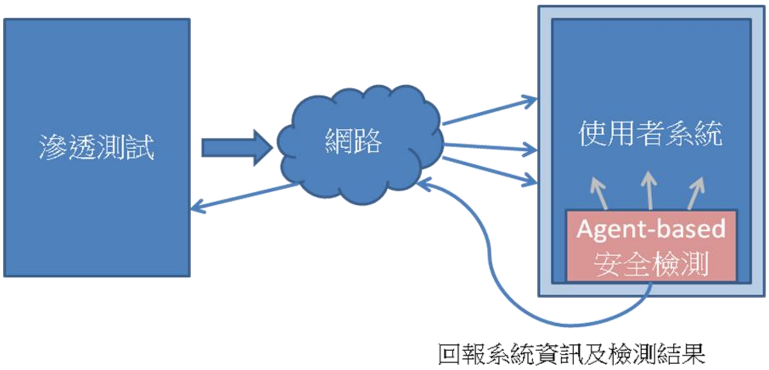


圖 18. Agent-based 合作式滲透測試系統架構

基於上述滲透測試的限制，本計畫將開發 Agent-based 合作式滲透測系統，其架構圖如圖 18 所示。利用在使用者系統上執行代理程式（Agent），從系統內部進行安全檢測，同時並與外部的滲透測試系統合作來協助滲透攻擊的進行。藉此我們將可進一步加強滲透測試的準確度，並彌補滲透測試廣度以及深度的不足。

本計畫將開發與代理程式合作的滲透測試系統，以使用者系統端執行的代理程式為主，在使用者授權的情況下，利用系統內部資訊進行使用者系統的安全檢測，掃描可能的安全漏洞，並且利用使用者系統目前執行的程序清單以及目前系統所開啟的網路通訊埠清單，回報給遠端的滲透測試系統以提供更精準的滲透攻擊。

◆ 系統內部安全檢測

利用在使用者系統上執行的代理程式，能夠偵測系統的內部資訊，包括：作業系統的種類及版本、目前執行的程序清單、目前開啟的網路通訊埠清單等相關資訊。代理程式能夠利用這些系統資訊，進行使用者系統內部的安全檢測，告知使用者在特定的作業系統環境下，特定的執行程序可能隱藏的安全漏洞；代理程式也可以在使用者系統內部，進行使用者密碼的強度檢測，提供相關的密碼建議，協助使用者設定更安全的密碼。利用代理程式進行使用者系統內部的安全檢測，能夠彌補滲透測試廣度的不足，與外部的滲透測試系統相輔相成。

在黑箱的滲透測試中，首先需要掃描發掘使用者系統的網路資訊，才能進一步地去偵測作業系統資訊，進行相關的滲透攻擊。滲透測試從使用者系統外部進行掃描，偵測使用

者系統開啟的網路通訊埠，進而推測使用者系統可能提供的服務或是執行的程式，並且發動相對應的滲透攻擊。由於僅從外面進行掃描，缺乏內部的執行資訊，推測到的服務或是程式難保不會失準，而沒有發動正確的滲透攻擊，使得滲透測試的精準度下降。

利用在使用者系統上執行的代理程式，能夠偵測系統的內部資訊，包括：作業系統的種類及版本、目前執行的程序清單、目前開啟的網路通訊埠清單等相關資訊。代理程式回傳這些資訊，能夠幫助外部的滲透測試，利用系統資訊使其更加掌握使用者系統的狀況，進而發動更準確的滲透攻擊，提升滲透測試的精準度。

- **行動核心網路拓樸探索工具**

行動核心網路拓樸探索工具為一套發展於 Android 智慧型手機上之網路服務監控程式。此工具控制 Android 手機底層的 Linux 作業系統來發送 ICMP 封包，並利用 ICMP 封包的 reply message 取得本機端與目標主機之間路徑上的所有節點資訊，並蒐集用戶使用智慧型手機上網的流量資訊與壅塞狀態。藉由分析 ICMP reply message，此工具將路徑上所有的節點資訊與封包回應時間記錄成 xml 檔，回傳給後方的伺服器端進行進一步的網路拓樸與流量分析，作為 ISP 業者管理行動網路與改善使用者經驗的重要依據·

行動核心網路拓樸探索工具可分為五項主要元件：(1) 負責處理使用者介面的 UI Thread、(2) 負責路由探索排程的 TraceRoute Worker Thread (TRW Thread)、(3) 負責 JAVA 和 Linux native application 溝通的 libajpcap_jni.so、(4) 產生 ICMP 封包並監聽 ICMP reply message 的 pcap_servive、和 (5) packet capture library (pcap library)。

本工具架構如圖 19 所示，UI Thread 和 TRW Thread 是架設在 Android SDK 與 NDK Framework 之上，libajpcap_jni.so 、pcap_servive 和 pcap library 則是屬於 Linux 作業系統的原件。為了使 Andorid 智慧型手機能夠支援 pcap library，我們透過 root Android 智慧型手機的方式，將 libajpcap_jni.so、pcap_servive、與 pcap library 植入底層的 Linux 作業系統中，使 TRW Thread 能透過 libajpcap_jni.so 所提供的 JNI (Java Native Interface) 介面操作 pcap_servive 和 pcap library，來蒐集智慧型手機所在地之擁塞狀況與路由資訊。

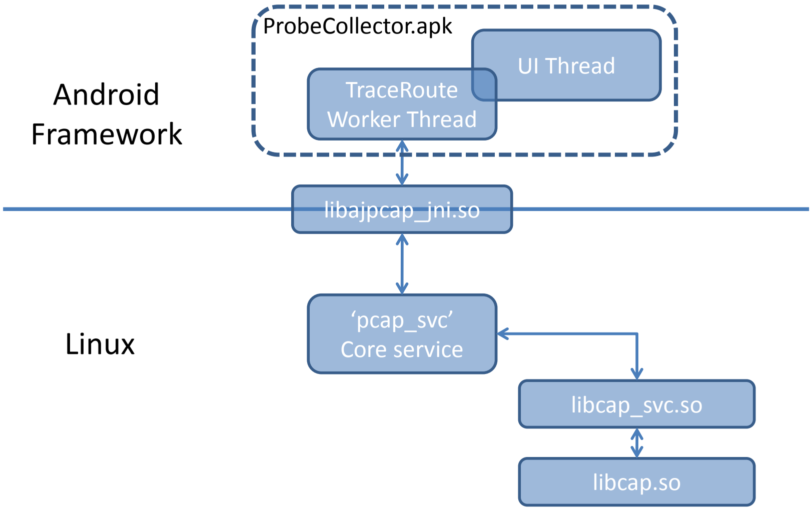


圖 19. 行動核心網路拓樸探索工具架構圖

以下分別介紹 本工具 之中每個元件的功能：

- UI Thread：

UI Thread 為 行動核心網路拓樸探索工具的使用者介面 (如圖 20 所示)，負責接收使用者輸入欲探索的 IP 地址，並將此 IP 地址交給 TRW Thread。TRW Thread 會將探索過後的結果傳給 UI Thread，由 UI Thread 顯示在智慧型手機的螢幕上。

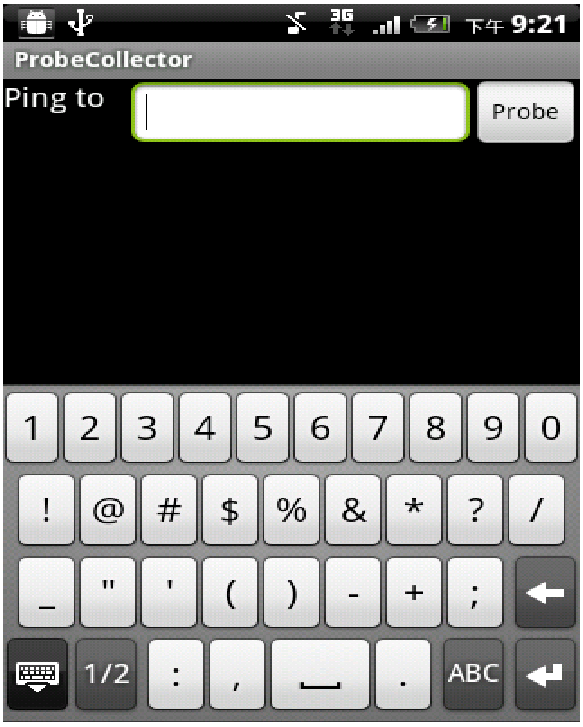
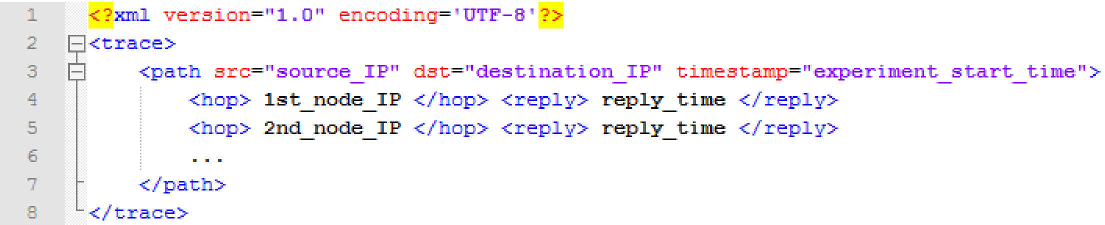


圖 20.行動核心網路拓樸探索工具的使用者介面

- TraceRoute Worker Thread (TRW Thread)：

TRW Thread 為一個小型的 TraceRoute 程式，負責路由探索的排程和對底層 Linux native application 的控制。TRW Thread 將使用者欲探索的 IP 地址和 TTL (Time-To-Live) 值透過 libajpcap_jni.so 的 JNI 介面傳給 pcap_servive，以操作 pcap_servive 發送 ICMP echo message 封包。TRW Thread 藉由不斷增加傳送給 pcap_servive 的 TTL 值，來達到探索封包傳遞路徑的目的。當收到目標 IP 位址的 ICMP reply message 後，TRW Thread 會將所收集到的節點回應時間結合透過 LocationManager 取得的經緯度資訊，依照探索到的節點順序存成如下圖 21 之 xml 檔，並交由 UI Thread 顯示和傳送到後端的 analysis 伺服器做進一步的分析。



```
<?xml version="1.0" encoding='UTF-8'?>
<trace>
    <path src="source_IP" dst="destination_IP" timestamp="experiment_start_time">
        <hop> 1st_node_IP </hop> <reply> reply_time </reply>
        <hop> 2nd_node_IP </hop> <reply> reply_time </reply>
        ...
    </path>
</trace>
```

圖 21.將蒐集到的 probe result 存成 xml 格式

◆ libajpcap_jni.so：

因 UI Thread 和 TRW Thread 主要是以 Java language 撰寫而成，並仰賴 Dalvik 虛擬機器執行轉譯過後的 bytecode。當 TRW Thread 需要與用 C language 寫的 pcap_servive 程式進行溝通時，Dalvik 虛擬機器會透過 libajpcap_jni.so 所提供的 JNI 介面，讓 TRW Thread 呼叫並執行 pcap_servive，並將執行時所需要的參數傳遞給 pcap_servive。當 pcap_servive 執行完成後，所取得的資料也透過 libajpcap_jni.so 回傳給 TRW Thread。

◆ pcap_servive：

pcap_servive 為一在開機時被 init.rc 啟動的背景服務，負責接收 TRW Thread 傳來的控制訊號 (IP 位址和 TTL 值)，並透過操作 pcap library 來發送 ICMP echo message 封包和處理 ICMP reply message。在傳送封包時，pcap_servive 略過正常的 protocol stack 的操作，直接填寫並建立 ICMP 封包的 header 與 content，並透過 pcap library 將 ICMP 的封包直接從 data link layer 傳送出去。在接收封包時，pcap_servive 透過 pcap library 監聽 data link layer 上收到的 raw packet。pcap_servive 利用 packet filter 過濾出所等待的 ICMP reply message，並分析 reply message 上的回應時間，傳給 TRW Thread 做進一步的判斷與處理。

◆ packet capture library (pcap library)

pcap library 由 libpcap.so 和 libpcap_svc.so 兩個 library 所組成，提供 pcap_servive 直接存取 data link layer 介面，來操作接收與發送 raw packet 的動作。

七、 計畫成果

在 2011 年我們的成果包含技術移轉 1 件、先期技術移轉 1 件、技術服務 1 件、與產學合作達 12 件，總金額超越本計畫書所規畫之 400 萬元。此項成果說明本計畫之建置成果在產業界之可應用性與前瞻性。本計畫之學術以及產業研究均有相當成果。參與本計畫之成員於 2011 年發表於國際重要期刊之論文數共 19 篇(多是 IEEE 以及 ACM 期刊)，發表於國際研討會之論文數共 16 篇以及國內研討會之論文共 2 篇。2011 年美國專利獲證 5 件，提出美國以及台灣專利申請各 1 件。本計畫的成果說明共分為三部分：（一）背景說明，（二）學術成果，（二）專利、技術創新與產學成果，以下將分別說明。

（一）背景說明

本計畫建置的檢測工具涵蓋了 Wired、WiFi、WiMAX 及 3.5G 多種異質網路環境，更支援兩種不同作業系統(Windows 及 Linux)，以提供多樣且完整的安全檢測服務。為了在 2011 年開發更適合產官學研的安全檢測工具，我們更與多個政府機關、法人以及產業界 (包括：總統府國家安全會議、法務部調查局、工研院、資策會、行政院研考會、國家資通安全會報技術服務中心、中華電信、友訊科技、宏達電、趨勢科技、喬鼎科技、中科院、教育部等) 合作並了解他們的檢測需求。舉例說明：由於行政院研考會、國家資通安全會報技術服務中心考慮推廣全國政府機關安全檢測評鑑，我們所開發的工具吻合部份他們的需求。例如：我們開發的 Agent-based 合作式滲透測試系統，客製化大規模遠端系統安全滲透檢測網以符合「windows 使用者電腦防護檢查」、「windows 伺服器主機防護檢查」和「Linux 伺服器主機防護檢查」，客製化網站伺服器安全滲透檢測系統以符合「網站安全性檢查表」。客製化惡意執行檔案檢測系統以及客製化動態惡意軟體行為分析檢測工具符合「防毒軟體防護」。而我們開發的線上 Windows 與 Linux 系統組態安全檢測系統也符合「組態設定安全」這個類別。此外，我們和工研院資通所已開始合作，將把本計畫所開發之資訊流動追蹤系統，應用至 ARM CPU 上進行軟體安全分析。此外，我們也和友訊科技及中華電信合作，進行軟體安全檢測以及惡意程式行為分析。這些單位在 Access Point/router、智慧型手機也有滲透檢測之需求。除了行動核心網路拓樸探索工具之外，中華電信也對於本計畫於 2011 年所開發之社群網路可疑連結隱藏內容發掘系統也有合作的意願。此外，我們也積極地與其他單位聯繫以及交流，來開發出更符合產研單位需求的安全檢測工具。本計畫主要的目的在於建置一個異質多網安全檢測平台、開發與建置產官學研所需的安全檢測工具，以及提供政府機關、產業界或是財團法人異質多網安全檢測的服務。

以下列出今年度所開發完成之 7 項全新的安全檢測工具，以及各工具對產業界及學術界之具體貢獻：

- 系統安全檢測類別
  - ■ 線上 Windows 與 Linux 系統組態安全檢測系統

    已與中華電信進行產學合作，預計未來將把此技術轉移至 Android 與中科院進行合作。
  - ■ Agent-based 合作式滲透測試系統

已與中華電信進行產學合作。

- ■ 行動核心網路拓樸探索工具

  已與宏達電 HTC 進行產學合作及先期技轉。

- ■ 作業系統 DNS 快取毒害監測與防禦工具

  已與中華電信研究所洽談，未來將進行合作。

- 軟體安全檢測類別
  - ■ 線上即時軟體行為分析檢測工具

    此項技術已與工研院進行技術轉移，並已與調查局、趨勢科技、喬鼎科技、中華電信進行產學合作。

- 人員安全意識安全檢測類別
  - ■ 社群網路可疑連結隱藏內容發掘系統

    已與中華電信進行產學合作。

  - ■ 行動平台漏洞模擬系統
    - 已與工研院進行產學合作。

為了普及並推廣我們所提供的安全檢測服務，我們將部份適切的檢測工具轉化成線上版本，並開始提供線上安全檢測服務於 TWISC@NCTU 網站上。

另外，我們努力的一個重點是希望能加強國際合作以持續推動改善台灣網路的整體安全架構。FIA (Future Internet Architecture)為本年度美國 NSF 的研發重點，其中又以安全為其核心重點，2011 年起三年共補助 30 million US dollars，總共通過四個大型研究計畫，其中通過最大的計劃為 UCLA 教授 Lixia Zhang 所領導， TWISC@NCTU 已經和該團對達成合作共識，彼此將加強技術交流，Professor Lixia Zhang 在 11 月也親自來台訪問 TWISC@NCTU。

對產業界之貢獻

藉由我們所建置的異質多網安全檢測平台與開發的安全檢測工具，我們希望提供政府機關、財團法人及高科技廠商無線網路安全檢測的服務，並且技轉我們所開發的檢測工具，以幫助上述單位發現漏洞及弱點。如此一來將可提高產業的經濟效益、提升無線產品附加價值、節省因網路攻擊或系統弱點所消耗的產值、節省專業檢測人力並且有效減少無線網路環境的攻擊。本計畫以建置開發為主，達到國內的資訊安全產業技術研發持續成長，達成資訊安全能量之提昇。持續結合國內各高科技廠商之資訊安全策略、管理、研發、教育等各方面人員共同參與，經由計畫之推動，結合各個來自國內各頂尖高科技廠商(例如：友訊科技、宏達電、趨勢科技等)的資訊安全技術領域的卓越成就共同研討交流，協助國內資訊安全產業研發能力與建立技術自主性、培育頂尖資訊安全科技研發人才，並藉由產學合作方式等促使人才實務化。

參與之工作人員獲得之工作訓練

參與本計畫的人員可在計畫執行過程學習到異質多網的基本概念及安全問題等相關的專業知識，並且可以透過工具的開發與實作來應用所學習到的專業知識。而學習到的知識及實務經驗將可以幫助成員未來跨入相關領域的職場有很大的幫助。除了專業知識，團隊也強調跨實驗室開發團隊互相合作的精神。參與的人員便可提前熟悉職場上工作團隊的合作模式，與同事

之間的工作互動就可更加的融洽，並提高工作效率。此外，計畫也積極的推動及促成產學合作(例如：與高科技廠商簽定 MOU)，透過計畫與高科技廠商和政府機關的合作模式，參與人員可以更瞭解目前廠商對科技的需求及如何規劃好的解決方法來符合需求。在參與過程更可瞭解業界和政府機關對產品開發的運作模式和人員的互動。相信這一方的寶貴經驗對參與人員在往後的職場上會有相當的幫助。因此，參與人員透過本計畫不但獲得了專業領域知識與實務經驗，更加獲得能與產業界及政府機關交流經驗，這對未來的人才的培育及提高國家競爭力有相當的幫助，並全面提昇工作人員在資訊安全上的品質與數量。

（二）學術成果

計畫主持人為謝續平老師，目前擔任 IEEE Trans. On Dependable and Secure Computing、IEEE Tran. On Reliability 編輯，IEEE RS Newsletter 總編輯，共同主持人為曾文貴老師、協同主持人為黃育綸、黃世昆、趙禧綠、吳育松、孫宏民等老師。參與學校包括國立交通大學與國立清華大學，參與實驗室共有 7 個。我們於交通大學電子資訊大樓內成立異質多網安全實驗室，提供跨校之師生計劃交流的平台。目前參與本計劃的碩博士生共計 33 位，其中博士生有 6 位。而在 2011 年本計劃共培育 22 位碩士生畢業，大部份的同學服務於相關企業中(例如：HTC、台積電及工研院等)，也有繼續留在國內或出國深造博士學位者。

參與本計畫之成員於 2011 年發表於國際重要期刊之論文數共 19 篇(多是 IEEE 以及 ACM 期刊)，發表於國際研討會之論文數共 16 篇以及國內研討會之論文共 2 篇(詳細列表如下所示)。

國際期刊論文

1. H.Y. Lin and W.G. Tzeng, “A Secure Erasure Code-based Cloud Storage System with Secure Data Forwarding,” *IEEE Transactions on Parallel and Distributed Systems, accepted for publication,* 2011.
2. C.L. Hou, C.C. Lu, S.C. Tsai, W.G. Tzeng, “An Optimal Data Hiding Scheme with Tree-Based Parity Check,” *IEEE Transactions on Image Processing, 20(3), pp.880-886.*
3. Y.L Huang, C.Y. Shen, S.P. Shieh, “S-AKA: A Provable and Secure Authentication Key Agreement Protocol for UMTS Networks,” *IEEE Transactions on Vehicular Technology, Vol, .60, No. 9, Noverber 2011, pp 4509 - 4519*
4. H.Y. Tsai, Y.L. Huang, “An Analytic Hierarchy Process-based Risk Assessment Method for Wireless Networks,” *IEEE Transactions on Reliability, (accepted), 2011.*
5. H.L. Chao, M.P. Hsu, “CTAP-Minimized Scheduling Algorithm for Millimeter Wave Based Wireless Personal Area Networks,” *IEEE Transactions on Vehicular Technology, Vol, .60, No. 8, October 2011, pp 3840 - 3852.*
6. C.F. Shih, W.J. Liao, H.L. Chao, “Exploiting Route Robustness in Joint Routing and Spectrum Allocation in Cognitive Radio Ad Hoc Networks,” *IEEE Transactions on Wireless Communications, accepted for publication ,2011.*
7. L.Y. Yeh, Y.L. Huang, S.P. Shieh, Anthony Joseph, “A Batch Authenticated and Key Agreement Framework for P2P-based Online Social Networks,” *IEEE Transactions on Vehicular Technology, accepted for publication, 2012*
8. C.W. Wang, Shiuhpyng Shieh,” SWIFT: Decoupling System-Wide Information Flow Tracking for Malware Analysis,” in revision, *IEEE Transactions on Reliability*.
9. C.W. Wang, Shiuhpyng Shieh, ”Detecting Privacy-Theft Android Apps with pTracker,” submitted to *IEEE Transactions on Information Forensics & Security*.

10. Hung-Min Sun*, W.-C. Ting, and K.-H. Wang, “On the Security of Chien's Ultra-lightweight RFID Authentication Protocol,”***IEEE Transactions on Dependable and Secure Computing***, Vol. 8, No. 2, pp. 315-317, 2011.

11. Hung-Min Sun*, H. Wang, K.-H. Wang, and C.-M. Chen, “A Native APIs Protection Mechanism in the Kernel Mode against Malicious Code,” ***IEEE Transactions on Computers***, Vol. 60. No. 6, pp. 813-823.

12. Hung-Min Sun* and B.-H. Ku, “Lossless Index Coding for Image Vector Quantization Using Huffman Codes,” *International Journal of Innovative Computing Information and Control,* accepted, to appear in Vol.7, No.10, October 2011. (SCI, Impact Factor: 1.664, Ranking: AUTOMATION & CONTROL SYSTEMS: top 12/60= 20%)

13. Hung-Min Sun*, B.-Z. He, S.-Y. Chang, and C.-H. Cho, “Efficient Authentication and Key Agreement Procedure in IP Multimedia Subsystem for UMTS,” *International Journal of Innovative Computing Information and Control,* accepted in 2011. (SCI, Impact Factor: 1.664, Ranking: AUTOMATION & CONTROL SYSTEMS: top 12/60= 20%)

14. S.-Y. Chang, Y.-H. Lin, Hung-Min Sun*, and M.-E. Wu, “Practical RSA Signature Scheme Based on Periodical Re-keying for Wireless Sensor Networks,” ***ACM Transactions on Sensor Networks***, accepted in 2010.

15. Hung-Min Sun*, C.-Y. Weng, S.-J. Wang*, C.-H. Yang, “Data Embedding in Image-media using Weight-function on Modulo Operations,” ***ACM Transactions on Embedded Computing Systems*, accepted in 2011.**

16. Hung-Min Sun*, C.-Y. Weng, C.-F. Lee, and C.-H. Yang, “Anti-forensics with Steganographic Data Embedding in Digital Images,” ***IEEE Journal on Selected Areas in Communications***, accepted in 2011.

17. C.-M. Chen, Y.-H. Lin, and Y.-C. Lin, and Hung-Min Sun*, “RCDA: Recoverable Concealed Data Aggregation for Data Integrity in Wireless Sensor Networks,” ***IEEE Transactions on Parallel and Distributed Systems***, accepted in 2011.

18. Y. Chen, J.-S. Chou, Hung-Min Sun, and M.-H. Cho, "A novel electronic cash system with trustee-based anonymity revocation from pairing," ***Electronic Commerce Research and Applications***, Vol. 10, No. 6, pp. 673-682, 2011.

19. Hung-Min Sun, Y.-H. Chen, and Y.-H. Lin, "oPass: A User Authentication Protocol Resistant to Password Stealing and Password Reuse Attacks," ***IEEE Transactions on Information Forensics and Security***, accepted in 2011.

國際會議論文

1. Y.R. Chen, Tygar, J.D, W.G. Tzeng, “Secure Group Key Management Using Uni-Directional Proxy Re-Encryption Schemes,” *In the 30th IEEE International Conference on Computer Communications (IEEE INFOCOM 2011)*
2. K.Y Chou, Y.R. Chen, W.G. Tzeng, “An Efficient and Secure Group Key Management Scheme Supporting Frequent Key Updates on Pay-TV systems,” *In the 13th Asia-Pacific Network Operations and Management Symposium (APNOMS 2011), September, 2011.*
3. H.Y. Lin, W.G. Tzeng, B.S. Lin, “A Decentralized Repair Mechanism for Decentralized Erasure Code based Storage Systems. In the 10th IEEE International Conference on Trust, “ *In the 10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (IEEE TrustCom-11).*
4. S.T. Shen, W.G. Tzeng, “Delegable Provable Data Possession for Remote Data in the Clouds,” *In the 13th International Conference on Information and Communications Security (ICICS*

2011).

5. H.L. Chao, T.C. Lou, S.H Jiang, “F2-MAC: an Efficient Channel Sensing and Access Mechanism for Cognitive Radio Networks,” ACM/Springer Wireless Networks, Vol. 17, Issue 5.
6. S.H Jiang, L.H. Chao, H.L. Chao, “A Decentralized MAC Protocol for Cognitive Radio Networks,” *(INFOCOM) 2011 Workshop on Cognitive & Cooperative Networks*.
7. S.F. Chou, H.L. Chao, C.L. Liu, “An Efficient Measurement Report Mechanism for Long Term Evolution Networks,” *(PIMRC) 2011*.
8. J.H. Liu, H.L. Chao, “Joint Relay Selection and Scheduling Algorithm for Inter-Piconet Communications in Millimeter Wave Wireless Personal Area Networks,” *(PIMRC) 2011*.
9. C.H. Chang, H.L. Chao, C.L. Liu, “Sum Throughput-Improved Resource Allocation with for LTE Uplink Transmission,” *(VTC) 2011 Fall, Sept. 2011*.
10. C.M. Fan, B.H. Li, S.P. Shieh, “On The Security of Password-Based Pairing Protocol in Bluetooth,” *The 13th Asia-Pacific Network Operations and Management Symposium, (APNOMS 2011), 2011*.
11. B.H. Li, S.P. Shieh, “RELEASE: Generating Exploits Using Loop-Aware Concolic Execution,” *IEEE Conference on Secure Software Integration and Reliability Improvement, June 2011*
12. Yun-Min Cheng, Bing-Han Li and Shiuhpyng Shieh, “Accelerating Taint-based Concolic Testing by Pruning Pointer Overtaint,” accepted for publication, IEEE Conference on Software Security and Reliability, Washington DC, June 2012.
13. Hung-Min Sun, C.-H. Chen, and P.-C. Li, “A Lightweight Secure Data Aggregation Protocol for Wireless Sensor Networks,” *2011 ICPP workshop on Applications of Wireless Ad Hoc and Sensor Networks*, 2011.
14. Hung-Min Sun, C.-S. Chen, C.-L. Chen, Y.-H. Chen, ”A Robust Defense Scheme to Resist Routing Attacks in Mobile Ad Hoc Networks,” *2011 3rd World Congress in Applied Computing, Computer Science, and Computer Engineering*, (*ACC 2011*), 2011.
15. Hung-Min Sun, C.-H. Chen, L.-C. Hsu, Y.-H. Chen, and Y.-H. Chen, “Reliable Data Transmission against Packet Dropping Misbehavior in Wireless Ad Hoc Networks,” *2011 IET International Communication Conference on Wireless Mobile and Computing, CCWMC 2011*, 2011.
16. Hung-Min Sun, C.-H. Chen, C.-W. Yeh, Y.-H. Chen, “A Collaborative Routing Protocol against Routing Disruptions in MANETs,” WCC-11 ACSA, 2011.

國內會議論文

1. Y.R. Liu, C.W. Wang, J.W. Hsu, S.P. Shieh, “Extracting Hidden Code from Packed Malware based on Virtual Machine Memory Comparison,” *21th Cryptology and Information Security Conference (CISC 2011), 2011*.
2. C.K. Chen, W.C. Chen, J.W. Hsu, S.P. Shieh, ” Mutant Malware Discovery and Behavior Analysis for Cyber Crime Investigation,” *22th Cryptology and Information Security Conference (CISC 2012), 2012.*

（三）專利、技術創新與產學成果

本計畫以異質多網環境的網路安全需求著手，與政府機關、財團法人、高科技廠商共同合作規劃。本年度我們在異質多網安全檢測平台下已開發與建置７種子系統與工具，這些工具已在實際運轉與測試中。以下列出目前申請的專利、本年度達成技術移轉、技術服務與產學合作清單。

專利

2011 年專利獲證 5 件

- S.I. Huang, S.P. Shieh, “Method and System for Secure Data Aggregation in Wireless Sensor Networks 無線傳感器網路中安全數據聚合的方法與系統,” China patent number ZL200710301500.9, 2011.
- S.I. Huang, S.P. Shieh, “Method and System for Secure Data Aggregation in Wireless Sensor Networks,” US patent no. 8027474, 2011.9.27.
- S.I. Huang, S.P. Shieh, “Method and System for Secure Data Aggregation in Wireless Sensor Networks, 用於在無線感應器網路中進行安全資料聚合的方法以及系統” ROC patent no. I350086. 10, 2011.
- Hung-Min Sun, Shih-Pu Hsu, and Chien-Ming Chen, “Mobile jamming attack method in wireless sensor network and method defending the same,” **US Patent 7,907,888**, March 15, 2011.
- Hung-Min Sun and Yue-Hsun Lin, “Pair-wise key pre-distribution method for wireless sensor network,” US Patent (pending), Application number: 11/599962, Publication number: US 20080044028/A1.

2011 我們提出專利申請共有 2 件，如下所示:

- 王繼偉，謝續平，劉晏如，“分離式的全系統層次模擬器與資訊流動追蹤方法與其應用,” Taiwan patent pending
- CW Wang, SP Shieh, YR Liu, “Method for Decoupling System-Wide Information Flow Tracking for Malware Analysis and Its Applications,” US patent pending

技術移轉【共 1 件】

- 建構於行動裝置 ARM CPU 上之污染分析系統，工業技術研究院

技術服務【共 1 件】

- 友訊科技 D-link－委託 Open D-Link Routers Forum 建置、維護與測試技術與諮詢服務 II；無線網路設備開放程式碼網站（社群）建置與安全性分析

產學合作【共 12 件，共 1674.4 萬】

- 調查局－法務部調查局惡意程式自動檢測技術支援系統委託研究採購案
- 教育部－DNSSEC 推動先期型計畫
- 中華電信－動態惡意程式行為側錄與汙染分析

- 中華電信－行動平台資通訊安全問題的研究(二)
- 中華電信－基於虛擬網路技術適用於異質網路之資源分配最佳化
- 中華電信－雙階層式全系統汙染鑑識分析
- 中華電信－行動平台資通訊安全問題的研究(三)
- 喬鼎科技－前瞻性檔案完整性驗證與可疑嵌入碼檢測平台
- 工研院－雲端行動的安全及時分析可行性評估先期探討
- 工研院－智慧終端技術研究
- 工研院－行動終端軟體品質技術研究
- 宏達電子(HTC)－雲端惡意程式鑑識與行動平台安全（含先期技轉）
- 教育部－DNSSEC 網域名稱安全架構建置與推廣計畫
- 趨勢科技--Technology Transfer on Network Threat Detection using Security Log Correlation

出國短期訪問報告書

撰寫時間： 2011 年　3 月　1　日

報告人： 交通大學謝續平

一、出國目的：

TWISC@NCTU（Taiwan Information Security Center at NCTU）執行國科會異質多網安全檢測平台建置計畫，出國參訪美國加州大學洛杉磯分校電機與資訊學系（Department of Electrical and Computer Engineering, UCLA)，國際交流。美國加州大學洛杉磯分校電機與資訊學系 Professor Lixia Zhang 是計算機網路與資訊安全界的權威，近年獲得美國國家科學基金會的 Future Internet Architecture (FIA)跨校的最大型計畫，為該項目所核准的四個大型計畫中規模最大者，該計畫的重點在於補足現有網際網路整體安全架構的不足，提出以 DNSSEC 來補強安全，藉由 DNSSEC 的佈建，提供了網際網路的全球資安基礎建設，進而可減低不明來源的各種網路與惡意軟體攻擊。這與本研究計畫高度相關，也與台灣網際網路整體安全息息相關，希望藉此機會瞭解下世代網際網路安全的發展規劃，獲得第一手的資訊。

二、參訪期間：

短期參訪期間為一百年一月二十六日至一百年二月十二日止，於

期間內赴美國加州大學洛杉磯分校電機與資訊學系（Department of Electrical and Computer Engineering, UCLA）國際合作短期參訪。詳細參訪行程如下：

1/25 抵達洛杉磯

1/26 – 2/12 visit Professor Lixia Zhang of UCLA

2/13 飛離洛杉磯

三、出國人員：

謝續平現任交通大學資訊工程系教授暨 TWISC@NCTU(Taiwan Information Security Center at NCTU)主任，總統府國家安全會議顧問，法務部調查局顧問。曾任交通大學資訊工程系系主任、交通大學計算機與網路中心主任、中華民國資訊安全學會理事長。謝教授積極參與國際活動，目前擔任 IEEE Reliability Society Ad Com 理事，Taipei/Tainan Chapter 主席，並榮獲榮譽獎項 ACM Distinguished Scientist（2010 年當年度全球 41 位得獎人之一）。在國際期刊編輯方面，現在擔任 IEEE Reliability Society Newsletter 總編輯、IEEE Tran. On Dependable and Secure Computing、IEEE Trans. On Reliability 副編輯，2012 IEEE Conference on Software Security and Reliability 議程主席。 曾任 ACM Tran. Information and System Security、Journal of Computer Security、JISE 副編輯、IEEE

Internet Computing 客座編輯。，曾創立並擔任 ACM Symposium on Information, Computer and Communications Security （ASIACCS）推動委員會主席（steering committee chair)，該會議為 ACM 旗艦會議，論文接受率僅約有 15%。

四、參訪經過及心得：

美國加州大學洛杉磯分校電機與資訊學系 Professor Lixia Zhang 是計算機網路與資訊安全界的權威，近年獲得美國國家科學基金會的 Future Internet Architecture (FIA)跨校的最大型計畫，該計畫的重點在於補足現有網際網路整體安全架構的不足，提出以 DNSSEC 來補強安全，進而建構下世代物流網與 Naming Networks，這與本研究計畫高度相關，也與台灣網際網路整體安全息息相關，

DNS（Domain Name Service)，為現今全世界最重要的服務之一，舉凡網路相關之事便離不開 DNS，即使是內部封閉網路，電腦間的溝通也需要透過 DNS 來解析。而 DNS 在最初設計時並沒有考慮到身分認證的功能，也造成了近年來 DNS 遭受攻擊的資安事件層出不窮，特別在 Kaminsky 發現 DNS 緩衝區漏洞攻擊後，DNS 安全性問題逐漸開始受到關注，追究其根本原因實為 DNS 協定本身之問題。

為了解決 DNS 本質上的不安全性，IETF 組織（Internet Engineering Task Force）定義了一套 DNSSEC 協定（Domain Name

System Security Extensions），可被視為 DNS 的安全性升級版。DNSSEC 導入了數位簽章的概念，能提供 DNS 資料驗證、資料完整性、資料存在性驗證等，藉此抵擋 Man-in-middle、DNS cache poisoning、DNS hijacking 等形式的攻擊，更進一步能安全的傳遞憑證等重要資料，應用到其他網路服務如電子郵件，提昇各種網路應用的安全性。

DNSSEC 協定在發展數年後的今日逐漸成熟並得到支持，國際組織與各國政府也開始著手進行部署。以目前的態勢來看，DNSSEC 已確定成為 DNS 的後繼標準，相關的國際標準制定已經成熟完備，不會再有太大的變動，各國政府與機構也積極進入 DNSSEC 佈署維運階段。

DNSSEC 的部署將被網路服務提供商視為未來網路基礎建設的重要目標，另一方面它在近年熱門的雲端運算(Cloud Computing)技術中也扮演著重要的角色，一旦所有服務都遷移到雲端，使用者獲得服務的首要步驟即是透過 DNS 服務轉譯，因此 DNS 的攻擊將造成服務中斷或惡意服務轉向。可見將傳統 DNS 升級為 DNSSEC，已成刻不容緩的任務。

經由 DNSSEC 的建構，可以形成全球可認證資料來源的網路，如此可以 DNSSEC 有兩方面的貢獻，第一，惡意程式的攻擊可因此而減輕。經由 DNSSEC 釣魚網站的身分可因此確認，進而降低網路駭客攻擊的威脅，也可減輕經由來路不明的網站或電子郵件的惡意程式攻

擊，換言之，來路不明的程式經由 DNSSEC 機制發現後，可立即傳送給惡意程式分析工具，能夠更快速有效率的達到偵測的目的；第二，基植於 DNSSEC 而建構的 Naming network 可確認物流網 Internet of Things 中 Things 的身分，進而未來下一世代物流網的安全性與實用性。Professor Zhang 與其學生多年從事 DNSSEC 的研究與發展，其學生在本學期畢業獲得博士學位後，也將加入 Verisign 從事 Global PKI 與 Malware behavior Analysis 的研究團隊，很高興在訪問的期間，經由多次研究討論，對於許多實際上研發可能碰到的問題能夠迅速的獲得解答，也因為瞭解這些 know how，也可以少走許多冤枉路，例如在甚麼狀況下 DNSSEC 會導致許多的電腦因設定不當而 crash，又例如 DNSSEC 的極限為何，先天上有何不足等等。這些研究的主題都是目前最熱門的研究課題，對本計畫未來的走向也有相當的助益。我們將以 DNSSEC、Phishing、Naming Network 為題撰寫一篇論文投稿至 IEEE Internet Computing。

為瞭解美國網路科技學術研發與技術發展現況，特別安排此次實地參訪活動，經由密集的實地參訪，深切的瞭解到美國對軟體技術的重視與投入，尤其在重點大學投資更是驚人，目前美國重點大學與產業之合作明顯超越台灣重點大學，且經由校內設立研究中心，使得學術研究已經與產業所需密切的結合，大學的研究真正的

達到提升產業技術的目的。

美國目前電腦網路頻寬已經遠超過台灣，網路安全的基礎建設也較台灣更為普遍，而全世界一流的高科技公司也紛紛在各重點大學設立研發中心，對產業技術發展具有催化作用，可以預見美國將在軟體技術持續保持領先。台灣過去教育頗為成功，創造了經濟奇蹟，現在因應大陸強力的競爭，台灣亦應大力投資軟體人才，厚植我國高科技基礎。

行政院國家科學委員會補助國內專家學者出席國際學術會議報告

101 年 5 月 28 日

報告人	陳柏廷	服務機構及職稱	國立交通大學電控工程研究所
會議時間地點	March 22-24, 2011 Hong Kong	本會核定補助編號	
會議名稱	2011 ACM Symposium on Information, Computer and Communications Security		
發表論文題目	Attacks Against Process Control Systems: Risk Assessment, Detection, and Response		

報告內容應包括下列事項：

一、參加會議經過：

本次大會共接受 35 篇 regular paper 和 24 篇 short paper，分為 13 個 Session 發表。大會另安排兩場 Keynote: (1) Prof. Jingguo Bi 主講的 ``Improved Nguyen-Vidick Heuristic Sieve Algorithm for Shortest Vector Problem” 探討改良的 Sieve Algorithm；(2) Prof. Gabriel Ghinita 主講的 ``Towards Mechanism for Detection and Prevention of Data Exfiltration by Insiders” 探討在 DBMS 層的 security 以期減少 Data Exfiltration 的問題。

二、與會心得：

此行最主要目的為發表`` Attacks Against Process Control Systems: Risk Assessment, Detection, and Response”論文一篇。此篇論文和傳統 IT 領域的安全研究略有不同，探討控制領域的安全性問題。從工業控制的角度來分析，當傳統控制走向電腦/網路控制時所可能遭遇的安全問題和可能的防護方式，因此獲得不少國外專家學者的建議。

三、考察參觀活動(無是項活動者省略)：

無

四、建議：

無

五、攜回資料名稱及內容：

ASIACCS 2011 國際研討會論文集，ISBN 978-145-030-564-8

Attacks Against Process Control Systems: Risk Assessment, Detection, and Response

Alvaro A. Cárdenas[§], Saurabh Amin[‡], Zong-Syun Lin[†],
Yu-Lun Huang[†], Chi-Yen Huang[†] and Shankar Sastry[‡]

[§] Fujitsu Laboratories of America
[‡] University of California, Berkeley
[†] National Chiao Tung University, Taiwan

ABSTRACT

In the last years there has been an increasing interest in the security of process control and SCADA systems. Furthermore, recent computer attacks such as the Stuxnet worm, have shown there are parties with the motivation and resources to effectively attack control systems.

While previous work has proposed new security mechanisms for control systems, few of them have explored new and fundamentally different research problems for securing control systems when compared to securing traditional information technology (IT) systems. In particular, the sophistication of new malware attacking control systems–malware including zero-days attacks, rootkits created for control systems, and software signed by trusted certificate authorities–has shown that it is very difficult to prevent and detect these attacks based solely on IT system information.

In this paper we show how, by incorporating knowledge of the physical system under control, we are able to detect computer attacks that change the behavior of the targeted control system. By using knowledge of the physical system we are able to focus on the final objective of the attack, and not on the particular mechanisms of how vulnerabilities are exploited, and how the attack is hidden. We analyze the security and safety of our mechanisms by exploring the effects of stealthy attacks, and by ensuring that automatic attack-response mechanisms will not drive the system to an unsafe state.

A secondary goal of this paper is to initiate the discussion between control and security practitioners–two areas that have had little interaction in the past. We believe that control engineers can leverage security engineering to design–based on a combination of their best practices–control algorithms that go beyond safety and fault tolerance, and include considerations to survive targeted attacks.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Network**]: Security and Protection; B.8.2 [**Performance and Reliability**]: Performance Analysis and Design Aids

ASIACCS '11, March 22–24, 2011, Hong Kong, China.

General Terms

Security

Keywords

SCADA, security, IDS, control systems, critical infrastructure protection, cyber-physical systems

1. INTRODUCTION

Control systems are computer-based systems that *monitor* and *control* physical processes. These systems represent a wide variety of networked information technology (IT) systems connected to the physical world. Depending on the application, these control systems are also called Process Control Systems (PCS), Supervisory Control and Data Aquisition (SCADA) systems (in industrial control or in the control of the critical infrastructures), Distributed Control Systems (DCS) or Cyber-Physical Systems (CPS) (to refer to embedded sensor and actuator networks).

Control systems are usually composed of a set of networked agents, consisting of sensors, actuators, control processing units such as programmable logic controllers (PLCs), and communication devices. For example, the oil and gas industry use integrated control systems to manage refining operations at plant sites, remotely monitor the pressure and flow of gas pipelines, and control the flow and pathways of gas transmission. Water utilities can remotely monitor well levels and control the wells pumps; monitor flows, tank levels, or pressure in storage tanks; monitor pH, turbidity, and chlorine residual; and control the addition of chemicals to the water.

Several control applications can be labeled as *safety-critical*: their failure can cause irreparable harm to the physical system being controlled and to the people who depend on it. SCADA systems, in particular, perform vital functions in national critical infrastructures, such as electric power distribution, oil and natural gas distribution, water and waste-water treatment, and transportation systems. They are also at the core of health-care devices, weapons systems, and transportation management. The disruption of these control systems could have a significant impact on public health, safety and lead to large economic losses.

Control systems have been at the core of critical infrastructures, manufacturing and industrial plants for decades, and yet, there have been few confirmed cases of cyberattacks. **Control systems, however, are now at a higher risk to computer attacks because their vulnerabilities are increasingly becoming exposed and available to an ever-growing set of motivated and highly-skilled attackers.**

No other attack demonstrates the threat to control systems as the

Stuxnet worm [1, 2]. The ultimate goal of Stuxnet is to sabotage that facility by reprogramming controllers to operate, most likely, out of their specified boundaries [1]. Stuxnet demonstrates that the motivation and capability exists for creating computer attacks capable to achieve military goals [3].

Not only can Stuxnet cause devastating consequences, but it is also very difficult to detect. Because Stuxnet used zero-day vulnerabilities, antivirus software would not have prevented the attack. In fact, the level of sophistication of the attack prevented some well known security companies such as Kaspersky to detect it initially [4]. In addition, victims attempting to detect modifications to their embedded controllers would not see any rogue code as Stuxnet hides its modifications with sophisticated PLC rootkits, and validated its drivers with trusted certificates.

The main motivation behind this work is the observation that while attackers may be able to hide the specific information technology methods used to exploit the system and reprogram their computers, they cannot hide their final goal: the need to cause an adverse effect on the physical system by sending malicious sensor or controller data that will not match the behavior expected by a supervisory control or an anomaly detection system.

Therefore, in this paper we explore security mechanisms that detect attacks by monitoring the physical system under control, and the sensor and actuator values. Our goal is to detect modifications to the sensed or controlled data as soon as possible, before the attack causes irreversible damages to the system (such as compromising safety margins).

In the rest of the paper we first summarize the vulnerability of control systems by discussing known attacks. We then discuss the efforts for securing control systems solely from an information technology perspective and identify the new and unique research problems that can be formulated by including a model of the physical system under control. We then develop a new attack detection algorithm and study the methodology on how to evaluate anomaly detection algorithms and their possible response strategies.

2. THE VULNERABILITY OF CONTROL SYSTEMS AND STUXNET

There have been many computer-based incidents in control systems. **Computer-based accidents** can be caused by any unanticipated software error, like the power plant shutdown caused by a computer rebooting after a patch [5]. **Non-targeted attacks** are incidents caused by the same attacks that any computer connected to the Internet may suffer, such as the Slammer worm infecting the Davis-Besse nuclear power plant [6], or the case of a controller being used to send spam in a water filtering plant [7].

However, the biggest threat to control systems are **Targeted attacks**. These attacks are the ones where the miscreants know that they are targeting control systems, and therefore, *they tailor their attack strategy with the aim of damaging the physical system under control*. Targeted attacks against control systems are not new. Physical attacks–for extortion and terrorism–are a reality in some countries [8]. Cyber-attacks are a natural progression to physical attacks: they are cheaper, less risky for the attacker, are not constrained by distance, and are easier to replicate and coordinate.

A classic computer-based targeted attack to SCADA systems is the attack on Maroochy Shire Council's sewage control system in Queensland, Australia [9]. There are many other reported targeted attacks [10–16]; however, no other attack has demonstrated the threats that control systems are subject to as well as the Stuxnet worm [1, 2]. Stuxnet has made clear that there are groups with the motivation and skills to mount sophisticated computer-based attacks to critical infrastructures, and that these attacks are not just speculations or belong only in Hollywood movies.

Stuxnet intercepts routines to read, write and locate blocks on a Programmable Logic Controller (PLC). By intercepting these requests, Stuxnet is able to modify the data sent to or returned from the PLC without the operator of the PLC ever realizing it [1].

Stuxnet was discovered on systems in Iran in June 2010 by researchers from Belarus–from the company VirusBlokAda; however, it is believed to have been released more than a year before. Stuxnet is a worm that spreads by infecting Windows computers. It uses multiple methods and zero-day exploits to spread itself via LANs or USB sticks. It is likely that propagation by LAN served as the first step, and propagation through removable drives was used to reach PCs not connected to other networks–therefore being isolated from the Internet or other networks is not a complete defense.

Once Stuxnet infects a Windows computer, it installs its own drivers. Because these drivers have to be signed, Stuxnet used two stolen certificates. Stuxnet also installs a rootkit to hide itself. The goal of the worm in a Windows computer is to search for WinCC/Step 7, a type of software used to program and monitor PLCs. (PLCs are the embedded systems attached to sensors and actuators that run control algorithms to keep the physical system operating correctly. They are typically programmed with a ladder logic program: a logic traditionally used to design control algorithms for panels of electromechanical relays.)

If Stuxnet does not find the WinCC/Step 7 software in the infected Windows machine, it does nothing; however, if it finds the software, it infects the PLC with another zero-day exploit, and then reprograms it. Stuxnet also attempts to hide the PLC changes with a PLC rootkit.

The reprogramming is done by changing only particular parts of the code–*overwriting certain process variables every five seconds and inserting rouge ladder logic*–therefore it is impossible to predict the effects of this change without knowing exactly how the PLC is originally programmed and what it is connected to, since the PLC program depends on the physical system under control, and typically, physical system parameters are unique to each individual facility. This means that the attackers were targeting a very specific PLC program and configuration (i.e., a very specific control system deployment).

Many security companies, including Symantec and Kaspersky have said that Stuxnet is the most sophisticated attack they have ever analyzed, and it is not difficult to see the reasons. Stuxnet uses four zero-day exploits, a Windows rootkit, the first known PLC rootkit, antivirus evasion techniques, peer-to-peer updates, and stolen certificates from trusted CAs. There is evidence that Stuxnet kept evolving since its initial deployment as attackers upgraded the infections with encryption and exploits, apparently adapting to conditions they found on the way to their target. The command and control architecture used two servers if the infected machines were able to access the Internet, or a peer to peer messaging system that could be used for machines that are offline. In addition, the attackers had a good level of intelligence about their target; they knew all the details of the control system configuration and its programs.

The sophistication of this attack has lead many to speculate that Stuxnet is the creation of a state-level sponsored attack. Since Iran has an unusually high percentage of the total number of reported infections of the worm in the world [1], there has been some speculation that their target was a specific industrial control system in Iran [2], such as a gas pipeline or power plant.

We argue that a threat like the Stuxnet worm must be dealt with defense-in-depth mechanisms like anomaly detection schemes. While traditional anomaly detection mechanisms may have some drawbacks like false alarms, we argue that for certain control systems, anomaly detection schemes focusing on the physical system–instead of using software or network models–can provide good detection capabilities with negligible false alarm rates.

3. NEW SECURITY PROBLEMS FOR CONTROL SYSTEMS

3.1 Efforts for Securing Control Systems

Most of the efforts for protecting control systems (and in particular SCADA) have focused on safety and reliability (the protection of the system against random and/or independent faults). Traditionally, control systems have not dealt with intentional actions or systematic failures. There is, however, an urgent growing concern for protecting control systems against malicious cyberattacks [6, 17–24].

There are several industrial and government-led efforts to improve the security of control systems. Several sectors–including chemical, oil and gas, and water–are currently developing programs for securing their infrastructure. The electric sector is leading the way with the North American Electric Reliability Corporation (NERC) cybersecurity standards for control systems [25]. NERC is authorized to enforce compliance to these standards, and it is expected that all electric utilities are fully compliant with these standards by the end of 2010.

NIST has also published a guideline for security best practices for general IT in Special Publication 800-53. Federal agencies must meet NIST SP800-53. To address the security of control systems, NIST has also published a Guide to Industrial Control System (ICS) Security [26], and a guideline to smart grid security in NIST-IR 7628. Although these recommendations are not enforceable, they can provide guidance for analyzing the security of most utility companies.

ISA (a society of industrial automation and control systems) is developing ISA-SP 99: a security standard to be used in manufacturing and general industrial controls.

The Department of Energy has also led security efforts by establishing the national SCADA test bed program [27] and by developing a 10-year outline for securing control systems in the energy sector [21]. The report–released in January 2006–identifies four main goals (in order from short-term goals to long-term goals): (1) measure the current security posture of the power grid, (2) develop and integrate protective measures, (3) implement attack detection and response strategies; and (4) sustain security improvements.

The use of wireless sensor networks in SCADA systems is becoming pervasive, and thus we also need to study their security. A number of companies have teamed up to bring sensor networks in the field of process control systems, and currently, there are two working groups to standardize their communications [28, 29]. Their wireless communication proposal has options to configure hop-by-hop and end-to-end confidentiality and integrity mechanisms. Similarly they provide the necessary protocols for access control and key management.

All these efforts have essentially three goals: (1) create awareness of security issues with control systems, (2) help control systems operators and IT security officers design a security policy, and (3) recommend basic security mechanisms for prevention (authentication, access controls, etc), detection, and response to security breaches.

While these recommendations and standards have placed significant importance on *survivability* of control systems (their ability to operate while they are under attack); we believe that they have not explored some new research problems that arise when control systems are under attack.

3.2 Differences

While it is clear that the security of control systems has become an active area in recent years, we believe that, so far, no one has been able to articulate what is new and fundamentally different in this field from a research point of view when compared to traditional IT security.

In this paper we would like to start this discussion by summarizing some previously identified differences and by proposing some new problems.

The property of control systems that is most commonly brought up as a distinction with IT security is that software **patching and frequent updates, are not well suited for control systems**. For example, upgrading a system may require months of advance in planning how to take the system offline; it is, therefore, economically difficult to justify suspending the operation of an industrial computer on a regular basis to install new security patches. Some security patches may even violate the certification of control systems, or–as previously mentioned–cause accidents to control systems [5].

Patching, however, is not a fundamental limitation to control systems. A number of companies have demonstrated that a careful antivirus and patching policy (e.g., the use of tiered approaches) can be used successfully [30, 31]. Also, most of the major control equipment vendors now offer guidance on both patch management and antivirus deployment for their control products. Thus there is little reason for SCADA system operators not to have good patch and antivirus programs in place today [32].

Large industrial control systems also have a large amount of **legacy systems**. Lightweight cryptographic mechanisms to ensure data integrity and confidentiality have been proposed to secure these systems [33, 34]. The recent IEEE P1711 standard is designed for providing security in legacy serial links [35]. Having some small level of security is better than having no security at all; however, *we believe that most of the efforts done for legacy systems should be considered as short-term solutions*. For properly securing critical control systems the underlying technology must satisfy some minimum performance requirements to allow the implementation of well tested security mechanisms and standards.

Another property of control systems that is commonly mentioned is the real-time requirements of control systems. Control systems are autonomous decision making agents which need to make decisions in real time. While availability is a well studied problem in information security, **real-time availability** provides a stricter operational environment than most traditional IT systems. We show in this paper that real-time availability requirements depend on the dynamics (fast vs. slow) of the physical system.

Not all operational differences are more severe in control systems than in traditional IT systems. By comparison to enterprise systems, control systems exhibit comparatively **simpler network dynamics**: Servers change rarely, there is a fixed topology, a stable user population, regular communication patterns, and a limited number of protocols. Therefore, implementing network intrusion detection systems, anomaly detection, and white listing may be easier than in traditional enterprise systems [36].

3.3 What is new and fundamentally different?

While all these differences are important, we believe that the major distinction of control systems with respect to other IT systems is the interaction of the control system with the physical world.

While current tools from information security can give *necessary* mechanisms for securing control systems, these mechanisms alone are not *sufficient* for defense-in-depth of control systems. When attackers bypass basic security defenses they may be able to affect the physical world.

In particular, research in computer security has focused traditionally on the protection of information; but it has not considered how attacks affect *estimation* and *control* algorithms–and ultimately, how attacks affect the physical world.

We believe that by understanding the interactions of the control system with the physical world, we should be able to develop a general and systematic framework for securing control systems in three fundamentally new areas:

1. Better understand the consequences of an attack for *risk assessment*. While there has been previous risk assessment studies on cyber security for SCADA systems [18, 37–39], currently, there are few studies on identifying the attack strategy of an adversary, once it has obtained unauthorized access to some control network devices. Notable exceptions are the study of false data-injection attacks to state estimation in power grids [40–45], and electricity markets [46]. We need further research to understand the threat model in order to design appropriate defenses and to invest in securing the most critical sensors or actuators.

2. Design new attack-detection algorithms. By monitoring the behavior of the physical system under control, we should be able to detect a wide range of attacks by compromised measurements. The work closest to ours are the study of false data injection attacks in control systems [47] and the intrusion detection models of Rrushi [48]–this last work; however, does not consider *dynamical models* of the process control system. We need further research on dynamical system models used in control theory as a tool for specification-based intrusion detection systems.

3. Design new attack-resilient algorithms and architectures: we need to design and operate control systems *to survive* an intentional cyber assault with no loss of critical functions. Our goal is to design systems where even if attackers manage to bypass some basic security mechanisms, they will still face several control-specific security devices that will minimize the damage done to the system. In particular, we need to investigate how to reconfigure and adapt control systems when they are under an attack to increase the resiliency of the system. We are not aware of any other work on designing new control algorithms or reconfiguration and control algorithms able to withstand attacks, or that reconfigure their operations based on detected attacks. There is previous work on safety and fault diagnosis; however, as we explain in this paper, these systems are not enough for detecting deception attacks launched by an intelligent attacker with knowledge on how to evade fault detection methods used by the system.

In the next sections we describe our ideas, experiments, and results for (1) risk-assessment, (2) false-data-injection detection, and (3) automatic attack-response in process control systems. In each section we first present a general theory for approaching the topic, and then for experimental validation, we implement our ideas to the model of a chemical reactor process.

4. RISK ASSESSMENT

Risk management is the process of shifting the odds in your favor by finding among all possible alternatives, the one that minimizes the impact of uncertain events.

Probably the best well known risk metric is the average loss $R_\mu = \mathbb{E}[L] \approx \sum_i L_i p_i$, where L_i is the loss if event i occurs, and p_i is the probability that event i occurs. Other risk metrics try to get more information about the probability distribution of the losses, and not only its mean value (R_μ). For example the variance of the losses $R_\chi = \mathbb{E}[L^2] - R_\mu$ is very useful in finance since it gives more information to risk averse individuals. This is particularly important if the average loss is computed for a large period of time (e.g. annually). If the loss is considered every time there is a computer event then we believe the average loss by itself provides enough risk information to make a rational decision.

In this paper we focus on attacks on sensor networks and the effects they have on the process control system. Therefore p_i denotes the likelihood that an attacker will compromise sensor i, and L_i denotes the losses associated with that particular compromise. To simplify our presentation we assume that p_i is the same for all sensors, therefore our focus in the remaining of this section is to estimate the potential losses L_i. The results can then be used to identify high priority sensors and to invest a given security budget in the most cost-effective way.

4.1 Attack models

We consider the case when the state of the system is measured by a sensor network of p sensors with measurement vector $y(k) = \{y_1(k), \ldots, y_p(k)\}$, where $y_i(k)$ denotes the measurement by sensor i at time k. All sensors have a dynamic range that defines the domain of y_i for all k. That is, all sensors have defined minimum and maximum values $\forall k, y_i(k) \in [y_i^{min}, y_i^{max}]$. Let $\mathcal{Y}_i = [y_i^{min}, y_i^{max}]$. We assume each sensor has a unique identity protected by a cryptographic key.

Let $\tilde{y}(k) \in \mathbb{R}^p$ denote the *received measurements by the controller* at time k. Based on these measurements the control system defines control actions to maintain certain operational goals. If some of the sensors are under attack, $\tilde{y}(k)$ may be different from the real measurement $y(k)$; however, we assume that the attacked signals $\tilde{y}_i(k)$ also lie within $\mathcal{Y}_i$ (signals outside this range can be easily detected by fault-tolerant algorithms).

Let $\mathcal{K}_a = \{k_s, \ldots, k_e\}$ represent the attack duration; between the start time k_s and stop time k_e of an attack. A general model for the observed signal is the following:

$$\tilde{y}_i(k) = \begin{cases} y_i(k) & \text{for } k \notin \mathcal{K}_a \\ a_i(k) & \text{for } k \in \mathcal{K}_a,\, a_i(k) \in \mathcal{Y}_i \end{cases}$$

where $a_i(k)$ is the attack signal. This general sensor attack model can be used to represent **integrity attacks** and **DoS attacks**. In an integrity attack we assume that if attackers have compromised a sensor, then they can inject any arbitrary value, therefore in this case, $a_i(k)$ is some arbitrary non-zero value.

In a DoS attack, the controller will notice the lack of new measurements and will react accordingly. An intuitive response for a controller to implement against a DoS attack is to use the last signal received: $a_i(k) = y_i(k_s)$, where $y_i(k_s)$ is the last measurement received before the DoS attack starts.

4.2 Experiments

To test our attacks, we use the Tennessee-Eastman process control system (TE-PCS) model and the associated multi-loop PI control law as proposed by Ricker [49]. We briefly describe the process architecture and the control loops in Figure 1. The original process model is implemented in FORTRAN and the PI control law is implemented in MATLAB. We use this code for our study.

The chemical process consists of an irreversible reaction which occurs in the vapour phase inside a reactor of fixed volume V of 122 (m^3). Two non-condensible reactants A and C react in the

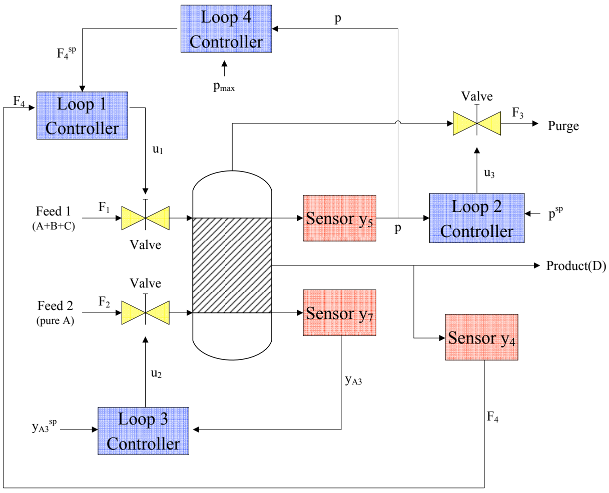


Figure 1: Architecture of the Simplified TE Plant.

presence of an inert B to form a non-volatile liquid product D:

$$A + C \xrightarrow{B} D.$$

The feed stream 1 contains A, C and trace of B; feed stream 2 is pure A; stream 3 is the purge containing vapours of A, B, C; and stream 4 is the exit for liquid product D. The measured flow rates of stream i is denoted by F_i $(kmol\ h^{-1})$. The *control objectives* are

- *Regulate* F_4, the rate of production of the product D, at a set-point F_4^{sp} $(kmol\ h^{-1})$,
- Maintain P, the operating pressure of the reactor, below the shut-down limit of 3000 kPa as dictated *safety* considerations,
- Minimize C, the *operating cost* measured in (kmol-of-product). The cost depends linearly on the purge loss of A and C relative to the production rate of D. The cost considerations dictate that the pressure be maintained as close as possible to 3000 kPa.

The production rate of D, denoted by r_D $(kmol\ h^{-1})$ is

$$r_D = k_0 y_{A3}^{v1} y_{C3}^{v2} P^{v3},$$

where y_{A3} and y_{C3} denote the respective fractions of A and C in the purge and v_1, v_2, v_3 are given constants.

There are four *input variables* (or command signals) available to achieve the above control objectives. The first three input variables, denoted as u_1, u_2 and u_3, trigger the actuators that can change the positions of the respective valves. The fourth input variable, denoted as u_4, is the set point for the proportional controller for the liquid inventory. The input variables as used by the controller in the following way:

- Production rate $y_4 = F_4$ is controlled using Feed 1 (u_1) by loop-1 controller,
- Pressure $y_5 = P$ is controlled using the purge rate (u_3) by loop-2 controller,
- Partial pressure of product A in the purge $y_7 = y_{A3}$ is controlled using Feed 2 (u_3) by loop-3 controller,

When u_3 saturates, the loop-4 controller uses u_1 to control the pressure P. The controllers for all four loops in figure 1 are *proportional integral* (PI) controllers.

In steady-state operation, the production rate F_4 is 100 $kmol\ h^{-1}$, the pressure P is 2700 KPa and the fraction of A in the purge is 47 $mol\%$.

We study the security issues of control systems by experimenting and simulating cyber attacks on sensor signals in the TE-PCS model. Because operating the chemical reactor with a pressure larger than 3000 kPa is unsafe (it may lead to an explosion or damage of the equipment) We.assume that that the goal of the attacker is to raise the pressure level of the tank to a value larger than 3000 kPa. We model an attacker that only has access to a single sensor at a given time. We also assume $L_i > L_j$, when an attack i can drive the system to an unsafe state and an attack j cannot, and $L_i = L_j$ if both attacks i and j either do not drive the system to an unsafe state, or both can compromise the safety of the sytem.

From the experimental results, we found that the most effective of these attacks were max/min attacks (i.e., when $a_i(k) = y_i^{min}$ or $a_i(k) = y_j^{max}$). However, not all of the max/min attacks were able to drive the pressure to unsafe levels. We now summarize some of the results.

- By attacking the sensors, a controller is expected to respond with incorrect control signals since it receives wrong information from the compromised sensors. For example, by forging y_7 as y_7^{max} from $t = 0$ to 30, the controller believes there is a large amount of component A in the tank.

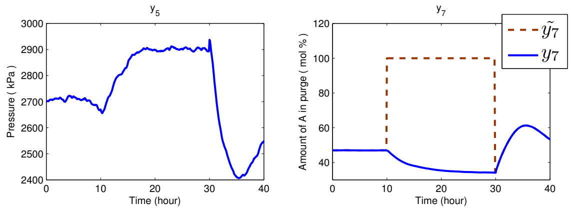
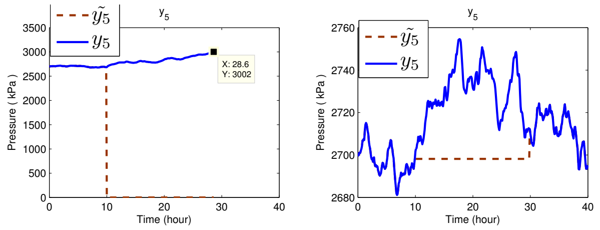


Figure 2: Integrity attack y_7^{max} from $t = 0$ to 30. The system remains in a safe state for attacks on y_7.

From the experiments, we found that the plant system can go back to the steady state after the attack finishes, as illustrated in Fig 2. Furthermore, the pressure in the main tank never reaches 3000 kPa. In general we found that the plant is very resilient to attacks on y_7 and y_4. Attacks in the limit of the sensing range (y^{min} and y^{max}) were the more damaging, but they did not force the system into an unsafe state.

- By launching attack y_5^{min} the controller turns down the purge valve to increase the pressure and prevent the liquid products from accumulating. We can see that the real pressure of the tank (y_5 in Fig 3(a)) keeps increasing past 3000 kPa and the system operates in an unsafe state. In this experiment, it takes about 20 hours ($t = 10$ to $t = 30$) to shut down (or cause an explosion to) the plant. This long delay in causing an effective attack may give defenders the advantage: for physical processes with *slow-dynamics*, it is possible that human system operators may have enough time to observe unusual phenomenon and take proper actions against the attack.
- We found out that in general DoS attacks do not affect the plant. We ran the plant 20 times for 40 hours each and for a DoS attack lasting 20 hours the pressure in the tank never exceeded 2900kPa.



9(a) 9(b)

Figure 3: Safety can be breached by compromising sensor y_5 (3(a)). DoS attacks, on the other hand, do not cause any damage (and they are easy to detect.) (3(b)).

We conclude that if the plant operator wants to prevent an attack from making the system operate in an unsafe state, it should prioritize defenses against integrity attacks rather than on DoS attacks. If the plant operator only has enough budget to deploy advanced security mechanisms for one sensor (e.g., tamper resistance, or TPM chips), y_5 should be the priority.

5. DETECTION OF ATTACKS

Detecting attacks to control systems can be formulated as an anomaly-based intrusion detection problem [50]. One big difference in control systems compared to traditional IT systems, is that instead of creating models of network traffic or software behavior, we can use a representative model of the physical system.

The intuition behind this approach is the following: if we know how the output sequence of the physical system, $y(k)$, should react to the control input sequence, $u(k)$, then any attack to the sensor data can be potentially detected by comparing the expected output $\hat{y}(k)$ with the received (and possibly compromised) signal $\tilde{y}(k)$. Depending on the quality of our estimate $\hat{y}(k)$ we may have some false alarms. We revisit this problem in the next section.

To formalize the anomaly detection problem, we need (1) a model of the behavior of the physical system, and (2) an anomaly detection algorithm. In section 5.1 we discuss our choice of linear models as an approximation of the behavior of the physical system. In section 5.2, we describe change detection theory and the detection algorithm we use–a nonparametric cumulative sum (CUSUM) statistic.

5.1 Linear Model

To develop accurate control algorithms, control engineers often construct a representative model that captures the behavior of the physical system in order to predict how the system will react to a given control signal. A process model can be derived from first principles (a model based on the fundamental laws of physics) or from empirical input and output data (a model obtained by simulating the process inputs with a carefully designed test sequence). It is also very common to use a combination of these two models; for example, first-principle models are typically calibrated by using process test data to estimate key parameters. Likewise, empirical models are often adjusted to account for known process physics [51, 52].

For highly safety-critical applications, such as the aerospace industry, it is technically and economically feasible to develop accurate models from first principles [51]. However, for the majority of process control systems, the development of process models from fundamental physics is difficult.

In many cases such detailed models are difficult to justify economically, and even impossible to obtain in reasonable time due to the complex nature of many systems and processes. (The TE-PCS system used in our experiments is one of the few cases available in the literature of a detailed nonlinear model of an industrial control problem; this is the reason why the TE-PCS system has been used as a standard testbed in many industrial control papers.)

To facilitate the creation of physical models, most industrial control vendors provide tools (called identification packages) to develop models of physical systems from training data. The most common models are *linear* systems. Linear systems can be used to model dynamics that are linear in state $x(k)$ and control input $u(k)$

$$x(k+1) = Ax(k) + Bu(k) \qquad (1)$$

where time is represented by $k \in \mathbb{Z}^+$, $x(k) = (x_1(k), \ldots, x_n(k)) \in \mathbb{R}^n$ is the state of the system, and $u(k) = (u_1(k), \ldots, u_m(k)) \in \mathbb{R}^m$ is the control input. The matrix $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ models the physical dependence of state i on state j, and $B = (b_{ij}) \in \mathbb{R}^{n\times m}$ is the input matrix for state i from control input j.

Assume the system (1) is monitored by a *sensor network* with p sensors. We obtain the measurement sequence from the observation equations

$$\hat{y}(k) = Cx(k), \qquad (2)$$

where $\hat{y}(k) = (\hat{y}_1(k), \ldots, \hat{y}_p(k)) \in \mathbb{R}^p$, and $\hat{y}_l(k) \in \mathbb{R}$ is the estimated measurement collected by sensor l at time k. Matrix $C \in \mathbb{R}^{p\times n}$ is called output matrix.

5.2 Detection Methods

The physical-model-based attack detection method presented in this paper can be viewed as complementary to intrusion detection methods based on network and computer systems models.

Because we need to detect anomalies in real time, we can use results from sequential detection theory to give a sound foundation to our approach. Sequential detection theory considers the problem where the measurement time is not fixed, but can be chosen online as and when the measurements are obtained. Such problem formulations are called *optimal stopping problems*. Two such problem formulations are: sequential detection (also known as sequential hypothesis testing), and quickest detection (also known as change detection). A good survey of these problems is given by Kailath and Poor [53].

In optimal stopping problems, we are given a time series sequence $z(1), z(2), \ldots, z(N)$, and the goal is to determine the minimum number of samples, N, the anomaly detection scheme should observe before making a decision d_N between two hypotheses: H_0 (normal behavior) and H_1 (attack).

The difference between sequential detection and change detection is that the former assumes the sequence $z(i)$ is generated either by the normal hypothesis (H_0), or by the attack hypothesis (H_1). The goal is to decide which hypothesis is true in minimum time. On the other hand, change detection assumes the observation $z(i)$ starts under H_0 and then, at a given k_s it changes to hypothesis H_1. Here the goal is to detect this change as soon as possible.

Both problem formulations are very popular, but security researchers have used sequential detection more frequently. However, for our attack detection method, the change detection formulation is more intuitive. To facilitate this intuition, we now briefly describe the two formulations.

5.2.1 Sequential Detection

Given a fixed probability of false alarm and a fixed probability of detection, the goal of sequential detection is to minimize the number of observations required to make a decision between two hypotheses. The solution is the classic sequential probability ratio test (SPRT) of Wald [54] (also referred as the threshold random walk (TRW) by some security papers). SPRT has been widely used in various problems in information security such as detecting portscans [55], worms [56], proxies used by spammers [57], and botnets [58].

Assuming that the observations $z(k)$ under H_j are generated with a probability distribution p_j, the SPRT algorithm can be described by the following equations:

$$S(k+1) = \log\frac{p_1(z(k))}{p_0(z(k))} + S(k)$$
$$N = \inf_n\{n : S(n) \notin [L, U]\},$$

starting with $S(0) = 0$. The SPRT decision rule d_N is defined as:

$$d_N = \begin{cases} H_1 & \text{if } S(N) \geq U \\ H_0 & \text{if } S(N) \leq L, \end{cases} \tag{3}$$

where $L \approx \ln\frac{b}{1-a}$ and $U \approx \ln\frac{1-b}{a}$, and where a is the desired probability of false alarm and b is the desired probability of missed detection (usually chosen as small values).

5.2.2 Change Detection

The goal of the change detection problem is to detect a possible change, at an unknown change point k_s.Cumulative sum (CUSUM) and Shiryaev-Roberts statistics are the two most commonly used algorithms for change detection problems. In this paper we use the CUSUM statistic because it is very similar to the SPRT.

Given a fixed false alarm rate, the CUSUM algorithm attempts to minimize the time N (where $N \geq k_s$) for which the test stops and decides that a change has occurred. Let $S(0) = 0$. The CUSUM statistic is updated according to

$$S(k+1) = \left(\log\frac{p_1(z(k))}{p_0(z(k))} + S(k)\right)^+ \tag{4}$$

where $(a)^+ = a$ if $a \geq 0$ and zero otherwise. The stopping time is:

$$N = \inf_n\{n : S(n) \geq \tau\} \tag{5}$$

for a given threshold τ selected based on the false alarm constraint.

We can see that the CUSUM algorithm is an SPRT test with $L = 0$, $U = \tau$, and whenever the statistic reaches the lower threshold L, it re-starts.

We now describe how to adapt the results of change detection theory to the particular problem of detecting compromised sensors. In the following, we use the subscript i to denote the sequence corresponding to sensor i.

One problem that we have in our case is that we do not know the probability distribution for an attack p_1. In general, an adaptive adversary can select any arbitrary (and possibly) non-stationary sequence $z_i(k)$. Assuming a fixed p_1 will thus limit our ability to detect a wide range of attacks.

To avoid making assumptions about the probability distribution of an attacker, we use ideas from nonparametric statistics. We do not assume a parametric distribution for p_1 and p_0; instead, only place mild constraints on the observation sequence. One of the simplest constraints is to assume the expected value of the random process $Z_i(k)$ that generates the sequence $z_i(k)$ under H_0 is less than zero ($\mathbb{E}_0[Z_i] < 0$) and the expected value of $Z_i(k)$ under H_1 is greater than zero ($\mathbb{E}_1[Z_i] > 0$).

To achieve these conditions let us define

$$z_i(k) := \|\tilde{y}_i(k) - \hat{y}_i(k)\| - b_i \tag{6}$$

where b_i is a small positive constant chosen such that

$$\mathbb{E}_0[\|\tilde{y}_i(k) - \hat{y}_i(k)\| - b_i] < 0. \tag{7}$$

The nonparametric CUSUM statistic for sensor i is then:

$$S_i(k) = (S_i(k-1) + z_i(k))^+,\ S_i(0) = 0 \tag{8}$$

and the corresponding decision rule is

$$d_{N,i} \equiv d_\tau(S_i(k)) = \begin{cases} H_1 & \text{if } S_i(k) > \tau_i \\ H_0 & \text{otherwise.} \end{cases} \tag{9}$$

where τ_i is the threshold selected based on the false alarm rate for sensor i.

Following [59], we state the following two important results for Eq. (8)-(9):

- The probability of false alarm decreases exponentially as the threshold τ_i increases,

- The time to detect an attack, $(N_i - k_{s,i})^+$, is inversely proportional to b_i.

5.3 Stealthy Attacks

A fundamental problem in intrusion detection is the existence of adaptive adversaries that will attempt to evade the detection scheme; therefore, we now consider an adversary that knows about our anomaly detection scheme. We take a conservative approach in our models by assuming a very powerful attacker with knowledge of: (1) the exact linear model that we use (i.e., matrices A,B, and C), the parameters (τ_i and b_i), and (3) the control command signals. Such a powerful attacker may be unrealistic in some scenarios, but we want to test the resiliency of our system to such an attacker to guarantee safety for a wide range of attack scenarios.

The goal of the attacker is to raise the pressure in the tank without being detected (i.e., raise the pressure while keeping the statistic he controls below the corresponding threshold τ_i).

We model three types of attacks: surge attacks, bias attacks and geometric attacks. Surge attacks model attackers that want to achieve maximum damage as soon as they get access to the system. A bias attack models attackers that try to modify the system discretely by adding small perturbations over a large period of time. Finally, geometric attacks model attackers that try to shift the behavior of the system very discretely at the beginning of the attack and then maximize the damage after the system has been moved to a more vulnerable state.

5.4 Surge Attacks

In a surge attack the adversary tries to maximize the damage as soon as possible, but when the statistic reaches the threshold, it then stays at the threshold level: $S_i(k) = \tau$ for the remaining time of the attack. To stay at the threshold, the attacker needs to solve the following quadratic equation:

$$S_i(k) + \sqrt{(\hat{y}_i(k) - \tilde{y}_i(k))^2} - b_i = \tau_i$$

The resulting attack (for y_5 and y_4) is:

$$\tilde{y}_i(k) = \begin{cases} y_i^{min} & \text{if } S_i(k+1) \leq \tau_i \\ \hat{y}_i(k) - |\tau_i + b_i - S_i(k)| & \text{if } S_i(k+1) > \tau_i \end{cases}$$

For y_7 we use

$$\tilde{y}_7(k) = \begin{cases} y_7^{max} & \text{if } S_{y_7}(k) \leq \tau_7 \\ \hat{y}_7 + |\tau_7 + b_7 - S_{y_7}(k)| & \text{if } S_{y_7}(k) > \tau_7 \end{cases}$$

5.5 Bias Attacks

In a bias attack the attacker adds a small constant c_i at each time step.

$$\tilde{y}_{i,k} = \hat{y}_{i,k} - c_i \in \mathcal{Y}_i$$

In this case, the nonparametric CUSUM statistic can be written as:

$$S_i(n) = \sum_{k=0}^{n-1} |\hat{y}_i(k) - \tilde{y}_i(k)| - nb_i$$

Assuming the attack starts at time $k = 0$ and assuming the attacker wants to be undetected for n time steps the attacker needs to solve the following equation:

$$\sum_{k=0}^{n-1} c_i = \tau_i + nb_i$$

Therefore $c_i = \tau_i/n + b$. This attack creates a bias of $\tau_i/n + b_i$ for each attacked signal.

This equation shows the limitations of the attacker. If an attacker wants to maximize the damage (maximize the bias of a signal), the attacker needs to select the smallest n it can find. Because $\tilde{y}_i \in \mathcal{Y}_i$ this attack reduces to an impulse attack.

If an attacker wants to attack for a long time, then n will be very large. If n is very large then the bias will be smaller.

5.6 Geometric Attacks

In a geometric attack, the attacker wants to drift the value very slowly at the beginning and maximize the damage at the end. This attack combines the slow initial drift of the bias attack with a surge attack at the end to cause maximum damage.

Let $\alpha \in (0, 1)$. The attack is:

$$\tilde{y}_i(k) = \hat{y}_i(k) - \beta_i \alpha_i^{n-k}.$$

Now we need to find α and β such that $S_i(n) = \tau_i$.

Assume the attack starts at time $k = 0$ and the attacker wants to be undetected for n time steps. The attacker then needs to solve the following equation.

$$\sum_{k=0}^{n-1} \beta_i \alpha_i^{n-k} - nb_i = \tau_i$$

This addition is a geometric progression.

$$\sum_{k=0}^{n-1} \beta_i \alpha_i^{n-k} = \beta_i \alpha_i^n \sum_{k=0}^{n-1} (\alpha_i^{-1})^k = \beta_i \frac{1 - \alpha_i^n}{\alpha_i^{-1} - 1}$$

By fixing α the attacker can select the appropriate β to satisfy the above equation.

5.7 Experiments

We continue our use of the TE-PCS model. In this section we first describe our selection criteria for matrices A, B, and C for the linear model, and the parameters b_i and τ_i for the CUSUM statistic. We then describe the tradeoffs between false alarm rates and the delay for detecting attacks. The section ends with the study of stealthy attacks.

5.7.1 Linear Model

In this paper we use the linear system characterized by the matrices A, B, and C, obtained by linearizing the non-linear TE-PCS model about the steady-state operating conditions. (See Ricker [49].) The linear model is a good representative of the actual TE-PCS model when the operating conditions are reasonably close to the steady-state.

5.7.2 Nonparametric CUSUM parameters

In order to select b_i for each sensor i, we need to estimate the expected value of the distance $|\hat{y}_i(k) - y_i(k)|$ between the linear model estimate $\hat{y}_i(k)$ and the sensor measurement $y_i(k)$ (i.e., the sensor signal without attacks).

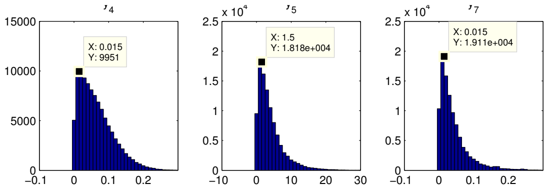


Figure 4: The paramenter of ADM: b. For y_4, 9951 bs are 0.015. The mean value of b_{y_4} is 0.0642.

We run experiments for ten thousand times (and for 40 hours each time) without any attacks to gather statistics. Fig 4 shows the estimated probability distributions (without normalization).

To obtain b_i, we compute the empirical expected value for each distance and then round up to the two most significant units. We obtain $b_{y_4} = 0.065$, $b_{y_5} = 4.1$, $b_{y_7} = 0.042$.

Once we have b_i for each sensor, we need to find a threshold τ_i to balance the tradeoff between false alarms and detection time.

False Alarm Rate.

We run simulations for twenty times without attacks and compute the total number of false alarms for different values of τ (and for each sensor). Fig 5 shows the results. Taking y_4 as an example, we notice that S_{y_4} alerts frequently if we set $\tau_{y_4} < 6$.

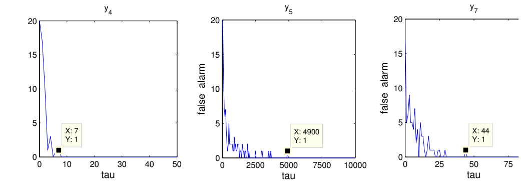


Figure 5: The number of false alarms decreases exponentially with increasing τ. This results confirm the theory supporting the nonparametric CUSUM algorithm.

In general, we would like to select τ as high as possible for each sensor to avoid any false alarm; however, increasing τ increases the time to detect attacks.

Detection Time.

To measure the time to detect attacks, we run simulations by launching scaling attacks ($a_i(k) = \lambda_m y_i(k)$) on sensors y_4, y_5 and y_7. Figs 6 and 7 shows the experimental results.

The selection of τ is a trade-off between detection time and the number of false alarms. The appropriate value differs from system to system. Because the large number of false alarms is one of the main problems for anomaly detection systems, and because the TE-PCS process takes at least 10 hours to reach the unsafe state (based on our risk assessment section), we choose the conservative set of parameters $\tau_{y_4} = 50$, $\tau_{y_5} = 10000$, $\tau_{y_7} = 200$. These parameters allow us to detect attacks within a couple of hours, while not raising any false alarms.

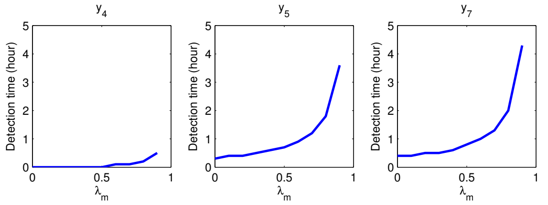


Figure 6: Detection time v.s. scaling attack. Note that for $\lambda_i^m = 1$ there is no alarm.

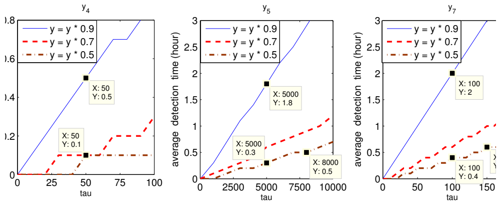


Figure 7: The time for detection increases linearly with increasing τ. This results confirm the theory behind the nonparametric CUSUM algorithm.

5.7.3 *Stealthy Attacks*

To test if our selected values for τ are resilient to stealthy attacks, we decided to investigate the effect of stealhty attacks as a function of τ. To test how the attacks change for all thresholds we parameterize each threshold by a parameter p: $\tau_i^{test} = p\tau_i$. Fig. 8 shows the percentage of times that geometric stealthy attacks (assuming the attacker controls all three sensor readings) were able to drive the pressure above 3000kPa while remaining undetected (as a function of p).

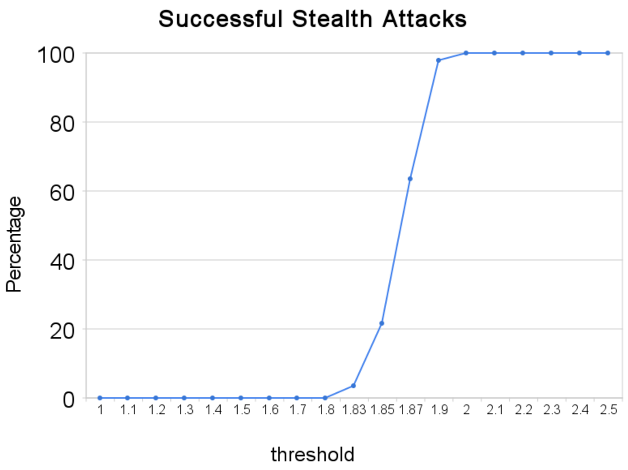


Figure 8: Percentage of stealthy attacks that increase the pressure of the tank above 3,000kPa as a function of scaling parameter p.

We implemented all stealth attacks starting at time $T = 10$ (hrs). We assume the goal of the attacker is to be undetected until $T = 30$ (hrs). For example, Fig. 9 shows the results of attacking all three sensors with a geometric attack. The nonparametric CUSUM statistic shown in Fig. 10 shows how the attacker remains undetected until time $T = 30$ (hrs).

We found that a surge attack does not cause significant damages because of the inertia of the chemical reactor: by the time the statistic reaches the threshold τ, the chemical reactor is only starting to respond to the attack. However, since the attacker can only add very small variations to the signal once it is close to the threshold, the attack ceases to produce any effect and the plant continues operating normally.

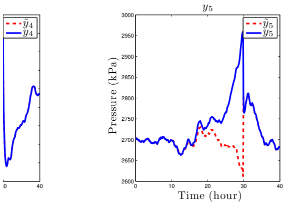


Figure 9: Geometric attacks to the three 3 sensors. The solid lines represent the real state of the system, while the dotted lines represent the information sent by the attacker.

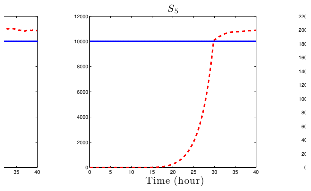


Figure 10: Statistics of geometric attacks with 3 sensors compromised.

Finally, we assume two types of attackers. An attacker that has compromised y_5 (but who does not know the values of the other sensors, and therefore can only control $S_{y_5}(k)$), and an attacker that has compromised all three sensors (and therefore can control the statistic $S(k)$ for all sensors). We launched each attack 20 times. The results are summarized in Figure 11.

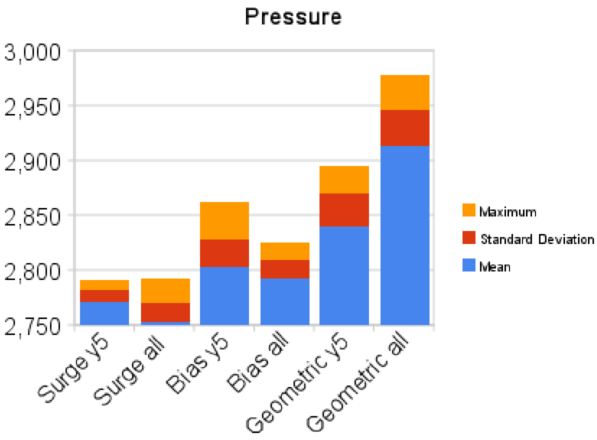


Figure 11: Effect of stealthy attacks. Each attack last 20 hours.

Our results show that even though our detection algorithm fails to detect stealthy attacks, we can keep the the plant in safe conditions. We also find that the most successful attack strategy are geometric attacks.

6. RESPONSE TO ATTACKS

A comprehensive security posture for any system should include mechanisms for prevention, detection, and response to attacks. Automatic response to computer attacks is one of the fundamental problems in information assurance. While most of the research efforts found in the literature focus on prevention (authentication, access controls, cryptography etc.) or detection (intrusion detection systems), in practice there are quite a few response mechanisms. For example, many web servers send CAPTCHAs to the client whenever they find that connections resemble bot connections, firewalls drop connections that conform to their rules, the execution of anomalous processes can be slowed down by intrusion detection systems, etc.

Given that we already have an estimate for the state of the system (given by a linear model), a natural response strategy for control systems is to use this estimate when the anomaly detection statistic fires an alarm. Fig 12 shows our proposed architecture. Specifically: for sensor i, if $S_i(k) > \tau_i$, the ADM replaces the sensor measurements $\tilde{y}_i(k)$ with measurements generated by the linear model $\hat{y}_i(k)$ (that is the controller will receive as input $\hat{y}_i(k)$ instead of $\tilde{y}_i(k)$). Otherwise, it treats $\tilde{y}_i(k)$ as the correct sensor signal.

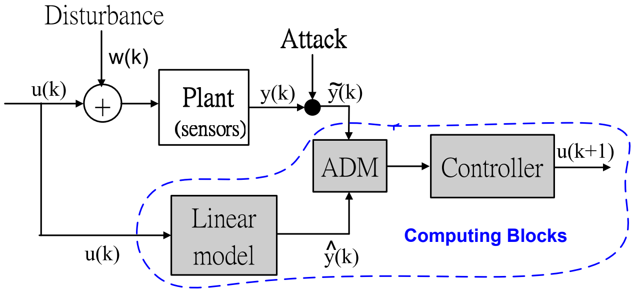
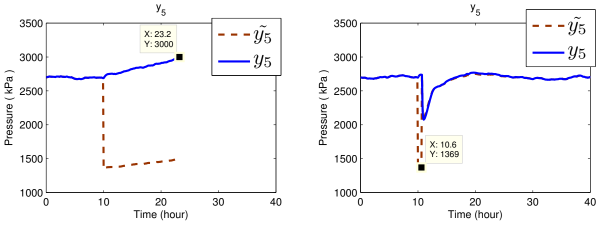


Figure 12: An Anomaly Detection Module (ADM) can detect an attack and send an estimate of the state of the system to the controller.

Introducing automatic response mechanisms is, however, not an easy solution. Every time systems introduce an automatic response to an alarm, they have to consider the cost of dealing with false alarms. In our proposed detection and response architecture (Fig. 12), we have to make sure that if there is a false alarm, controlling the system by using the estimated values from the linear system will not cause any safety concerns.

6.1 Experiments

The automatic response mechanism works well when we are under attack. For example, Fig. (13) shows that when an attack is detected, the response algorithm manages to keep the system in a safe state. Similar results were obtained for all detectable attacks.



9(a) Without ADM

9(b) ADM detects and responds to the attack at $T = 10.7$ (hr)

Figure 13: $\tilde{y_5} = y_5 * 0.5$

Alarms	Avg y_5	Std Dev	Max y_5
0	2700.4	14.73	2757

Table 1: For Thresholds $\tau_{y_4} = 50, \tau_{y_5} = 10000, \tau_{y_7} = 200$ we obtain no false alarm. Therefore we only report the expected pressure, the standard deviation of the pressure, and the maximum pressure reached under no false alarm.

While our attack response mechanism is a good solution when the alarms are indeed an indication of attacks, Our main concern in this section is the cost of false alarms. To address these concerns we ran the simulation scenario without any attacks 1000 times; each time the experiment ran for 40 hours. As expected, with the parameter set $\tau_{y_4} = 50$, $\tau_{y_5} = 10000$, $\tau_{y_7} = 200$ our system did not detect any false alarm (see Table 1); therefore we decided to reduce the detection threshold to $\tau_{y_4} = 5$, $\tau_{y_5} = 1000$, $\tau_{y_7} = 20$ and run the same experiments again. Table 2 shows the behavior of the pressure *after the response to a false alarm.* We can see that while a false response mechanism increases the pressure of the tank, it never reaches unsafe levels. The maximum pressure obtained while controlling the system based on the linear model was $2779 kPa$, which is in the same order of magnitude than the normal variation of the pressure without any false alarm ($2757 kPa$).

In our case, even if the system is kept in a safe state by the automated response, our response strategy is meant as a temporary solution before a human operator responds to the alarm. Based on our results we believe that the time for a human response can be very large (a couple of hours).

7. CONCLUSIONS

In this work we identified three new research challenges for securing control systems. We showed that by incorporating a physical model of the system we were able to identify the most critical sensors and attacks. We also studied the use of physical models for anomaly detection and proposed three generic types of stealthy attacks. Finally, we proposed the use of automatic response mechanisms based on estimates of the state of the system. Automatic responses may be problematic in some cases (especially if the response to a false alarm is costly); therefore, we would like to emphasize that the automatic response mechanism should be considered as a temporary solution before a human investigates the alarm. A full deployment of any automatic response mechanism should take into consideration the amount of time in which it is reasonable for a human operator to respond, and the potential side effects of

	Alarms	Avg y_5	Std Dev	Max y_5
y_4	61	2710	30.36	2779
y_5	106	2705	18.72	2794
y_7	53	2706	20.89	2776

Table 2: Behavior of the plant after response to a false alarm with thresholds $\tau_{y_4} = 5, \tau_{y_5} = 1000, \tau_{y_7} = 20$.

responding to a false alarm.

In our experiments with the TE-PCS process we found several interesting results. (1) Protecting against integrity attacks is more important than protecting against DoS attacks. In fact, we believe that DoS attacks have negligible impact to the TE-PCS process. (2) The chemical reactor process is a well-behaved system, in the sense that even under perturbations, the response of the system follows very closely our linear models. In addition, the slow dynamics of this process allows us to be able to detect attacks even with large delays with the benefit of not raising any false alarms. (3) Even when we configure the system to have false alarms, we saw that the automatic response mechanism was able to control the system in a safe mode.

One of our main conclusions regarding the TE-PCS plant, is that it is a very resiliently-designed process control system. Design of resilient process control systems takes control system design experience and expertise. The design process is based on iteratively evaluating the performance on a set of bad situations that can arise during the operation of the plant and modifying control loop structures to build in resilience. In particular, Ricker's paper discusses the set of random faults that the four loop PI control is able to withstand.

We like to make two points in this regard: (1). The PI control loop structure is distributed, in the sense that no PI control loop controls all actuators and no PI loop has access to all sensor measurements, and (2). The set of bad situations to which this control structure is able to withstand may itself result from the one or more cyber attacks. However, even though the resilience of TE-PCS plant is ensured by expert design, we find it interesting to directly test this resilience within the framework of assessment, detection and response that we present in this article.

However, as a word of caution, large scale control system designs are often not to resilient by design and may become prey to such stealth attacks if sufficient resilience is not built by design in the first place. Thus, our ideas become all the more relevant for operational security until there is a principled way of designing fully attack resilient control structures and algorithms (which by itself is a very challenging research endeavor and may not offer a cost effective design solution).

Even though we have focused on the analysis of a chemical reactor system, our principles and techniques can be applied to many other physical processes. An automatic detection and response module may not be a practical solution for all control system processes; however, we believe that many processes with similar characteristics to the TE-PCS can benefit from this kind of response.

Acknowledgments

We would like to thank Gabor Karsai, Adrian Perrig, Bruno Sinopoli, and Jon Wiley for helpful discussions on the security of control systems. This work was supported in part by by the iCAST-TRUST collaboration project, and by CHESS at UC Berkeley, which receives support from the NSF awards #0720882 (CSR-EHS: PRET) and #0931843 (ActionWebs), ARO #W911NF-07-2-0019, MURI #FA9550-06-0312, AFRL, and MuSyC.

8. REFERENCES

[1] Nicolas Falliere, Liam O Murchu, and Eric Chien. *W32.Stuxnet Dossier*. Symantec, version 1.3 edition, November 2010.

[2] Ralph Langner. Langner communications. `http://www.langner.com/en/`, October 2010.

[3] Steve Bellovin. Stuxnet: The first weaponized software? `http://www.cs.columbia.edu/~smb/blog//2010-09-27.html`, October 2010.

[4] Dale Peterson. Digital bond: Weisscon and stuxnet. `http://www.digitalbond.com/index.php/2010/09/22/weisscon-and-stuxnet/`, October 2010.

[5] Brian Krebs. *Cyber Incident Blamed for Nuclear Power Plant Shutdown*. Washington Post, `http://www.washingtonpost.com/wp-dyn/content/article/2008/06/05/AR2008060501958.html`, June 2008.

[6] Robert J. Turk. Cyber incidents involving control systems. Technical Report INL/EXT-05-00671, Idao National Laboratory, October 2005.

[7] Richard Esposito. Hackers penetrate water system computers. `http://blogs.abcnews.com/theblotter/2006/10/hackers_penetra.html`, October 2006.

[8] BBC News. *Colombia Rebels Blast Power Pylons*. BBC, `http://news.bbc.co.uk/2/hi/americas/607782.stm`, January 2000.

[9] Jill Slay and Michael Miller. Lessons learned from the maroochy water breach. In *Critical Infrastructure Protection*, volume 253/2007, pages 73–82. Springer Boston, November 2007.

[10] Paul Quinn-Judge. Cracks in the system. *TIME Magazine*, 9th Jan 2002.

[11] Thomas Reed. *At the Abyss: An Insider's History of the Cold War*. Presidio Press, March 2004.

[12] United States Attorney, Eastern District of California. Willows man arrested for hacking into Tehama Colusa Canal Authority computer system. `http://www.usdoj.gov/usao/cae/press_releases/docs/2007/11-28-07KeehnInd.pdf`, November 2007.

[13] United States Attorney, Eastern District of California. Sacramento man pleads guilty to attempting ot shut down california's power grid. `http://www.usdoj.gov/usao/cae/press_releases/docs/2007/12-14-07DenisonPlea.pdf`, November 2007.

[14] David Kravets. Feds: Hacker disabled offshore oil platform leak-detection system. `http://www.wired.com/threatlevel/2009/03/feds-hacker-dis/`, March 2009.

[15] John Leyden. Polish teen derails tram after hacking train network. *The Register*, 11th Jan 2008.

[16] Andrew Greenberg. Hackers cut cities' power. In *Forbes*, Jaunuary 2008.

[17] V.M. Igure, S.A. Laughter, and R.D. Williams. Security issues in SCADA networks. *Computers & Security*, 25(7):498–506, 2006.

[18] P. Oman, E. Schweitzer, and D. Frincke. Concerns about intrusions into remotely accessible substation controllers and SCADA systems. In *Proceedings of the Twenty-Seventh Annual Western Protective Relay Conference*, volume 160. Citeseer, 2000.

[19] US-CERT. *Control Systems Security Program*. US Department of Homeland Security, `http://www.us-cert.gov/control_systems/index.html`, 2008.

[20] GAO. Critical infrastructure protection. Multiple efforts to secure control systems are under way, but challenges remain. Technical Report GAO-07-1036, Report to Congressional Requesters, September 2007.

[21] Jack Eisenhauer, Paget Donnelly, Mark Ellis, and Michael O'Brien. *Roadmap to Secure Control Systems in the Energy Sector*. Energetics Incorporated. Sponsored by the U.S. Department of Energy and the U.S. Department of

Homeland Security, January 2006.

[22] Eric Byres and Justin Lowe. The myths and facts behind cyber security risks for industrial control systems. In *Proceedings of the VDE Congress, VDE Association for Electrical Electronic & Information Technologies*, October 2004.

[23] D. Geer. Security of critical control systems sparks concern. *Computer*, 39(1):20–23, Jan. 2006.

[24] A.A. Cardenas, T. Roosta, and S. Sastry. Rethinking security properties, threat models, and the design space in sensor networks: A case study in SCADA systems. *Ad Hoc Networks*, 2009.

[25] NERC-CIP. *Critical Infrastructure Protection*. North American Electric Reliability Corporation, `http://www.nerc.com/cip.html`, 2008.

[26] K. Stouffer, J. Falco, and K. Kent. Guide to supervisory control and data acquisition (SCADA) and industrial control systems security. Sp800-82, NIST, September 2006.

[27] Idaho National Laboratory. National SCADA Test Bed Program. `http://www.inl.gov/scada`.

[28] Hart. `http://www.hartcomm2.org/frontpage/wirelesshart.html`. *WirelessHart whitepaper*, 2007.

[29] ISA. `http://isa.org/isasp100`. *Wireless Systems for Automation*, 2007.

[30] Eric Cosman. Patch management at Dow chemical. In *ARC Tenth Annual Forum on Manufacturing*, February 20-24 2006.

[31] Patch management strategies for the electric sector. Edison Electric Institute–IT Security Working Group, March 2004.

[32] Eric Byres, David Leversage, and Nate Kube. Security incidents and trends in SCADA and process industries. *The Industrial Ethernet Book*, 39(2):12–20, May 2007.

[33] Andrew K. Wright, John A. Kinast, and Joe McCarty. Low-latency cryptographic protection for SCADA communications. In *Applied Cryptography and Network Security (ACNS)*, pages 263–277, 2004.

[34] Patrick P. Tsang and Sean W. Smith. YASIR: A low-latency high-integrity security retrofit for lecacy SCADA systems. In *23rd International Information Security Conference (IFIC SEC)*, pages 445–459, September 2008.

[35] Steven Hurd, Rhett Smith, and Garrett Leischner. Tutorial: Security in electric utility control systems. In *61st Annual Conference for Protective Relay Engineers*, pages 304–309, April 2008.

[36] Steven Cheung, Bruno Dutertre, Martin Fong, Ulf Lindqvist, Keith Skinner, and Alfonso Valdes. Using model-based intrusion detection for SCADA networks. In *Proceedings of the SCADA Security Scientific Symposium*, Miami Beach, FL, USA, 2007 2007.

[37] PAS Ralston, JH Graham, and JL Hieb. Cyber security risk assessment for SCADA and DCS networks. *ISA transactions*, 46(4):583–594, 2007.

[38] P.A. Craig, J. Mortensen, and J.E. Dagle. Metrics for the National SCADA Test Bed Program. Technical report, PNNL-18031, Pacific Northwest National Laboratory (PNNL), Richland, WA (US), 2008.

[39] G. Hamoud, R.L. Chen, and I. Bradley. Risk assessment of power systems SCADA. In *IEEE Power Engineering Society General Meeting, 2003*, volume 2, 2003.

[40] Yao Liu, Michael K. Reiter, and Peng Ning. False data injection attacks against state estimation in electric power grids. In *CCS '09: Proceedings of the 16th ACM conference on Computer and communications security*, pages 21–32, New York, NY, USA, 2009. ACM.

[41] Rakesh Bobba, Katherine M. Rogers, Qiyan Wang, Himanshu Khurana, Klara Nahrstedt, and Thomas J. Overbye. Detecting false data injection attacks on dc state estimation. In *Preprints of the 1st Workshop on Secure Control Systems*, 2010.

[42] Henrik Sandberg, Teixeira Andre, and Karl H. Johansson. On security indices for state estimators in power networks. In *Preprints of the 1st Workshop on Secure Control Systems*, 2010.

[43] Oliver Kosut, Liyan Jia, Robert J. Thomas, and Lang Tong. Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures. In *First International Conference on Smart Grid Communications (SmartGridComm)*, pages 220–225, 2010.

[44] Oliver Kosut, Liyan Jia, Robert J. Thomas, and Lang Tong. On malicious data attacks on power system state estimation. In *UPEC*, 2010.

[45] A Teixeira, S. Amin, H. Sandberg, K.H. Johansson, and S.S. Sastry. Cyber-security analysis of state estimators in electric power systems. In *IEEE Conference on Decision and Control (CDC)*, 2010.

[46] Le Xie, Yilin Mo, and Bruno Sinopoli. False data injection attacks in electricity markets. In *First International Conference on Smart Grid Communications (SmartGridComm)*, pages 226–231, 2010.

[47] Yilin Mo and Bruno Sinopoli. False data injection attacks in control systems. In *Preprints of the 1st Workshop on Secure Control Systems*, 2010.

[48] Julian Rrushi. *Composite Intrusion Detection in Process Control Networks*. PhD thesis, Universita Degli Studi Di Milano, 2009.

[49] NL Ricker. Model predictive control of a continuous, nonlinear, two-phase reactor. *JOURNAL OF PROCESS CONTROL*, 3:109–109, 1993.

[50] Dorothy Denning. An intrusion-detection model. *Software Engineering, IEEE Transactions on*, SE-13(2):222–232, Feb. 1987.

[51] S. Joe Quin and Thomas A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, July 2003.

[52] J.B. Rawlings. Tutorial overview of model predictive control. *Control Systems Magazine, IEEE*, 20(3):38–52, Jun 2000.

[53] T. Kailath and H. V. Poor. Detection of stochastic processes. *IEEE Transactions on Information Theory*, 44(6):2230–2258, October 1998.

[54] A. Wald. *Sequential Analysis*. J. Wiley & Sons, New York, 1947.

[55] Jaeyeon Jung, Vern Paxson, Arthur Berger, and Hari Balakrishan. Fast portscan detection using sequential hypothesis testing. In *Proceedings of the 2004 IEEE Symposium on Security and Privacy*, pages 211–225, May 2004.

[56] Stuart Schechter and Jaeyeon Jung Arthur Berger. Fast detection of scanning worm infections. In *Proc. of the Seventh International Symposium on Recent Advances in Intrusion Detection (RAID)*, September 2004.

[57] M. Xie, H. Yin, and H. Wang. An effective defense against email spam laundering. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pages 179–190, October 30–November 3 2006.

[58] Guofei Gu, Junjie Zhang, and Wenke Lee. Botsniffer: Detecting botnet command and control channels in network traffic. In *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS'08)*, San Diego, CA, February 2008.

[59] B.E. Brodsky and B.S. Darkhovsky. *Non-Parametric Methods in Change-Point Problems*. Kluwer Academic Publishers, 1993.

行政院國家科學委員會補助國內專家學者出席國際學術會議報告

100 年 11 月 30 日

報告人姓名	沈宣佐	服務機構 及職稱	國立交通大學資訊科學與工程研究所
時間 會議 地點	Nov 23 – 26, 2011 Beijing, China	本會核定 補助文號	
會議 名稱	(中文) 第 13 屆資訊與通訊安全國際研討會 (英文) The 13th International Conference on Information and Communications Security		
發 表 論 文 題目	Delegable Provable Data Possession for Remote Data in the Clouds		

報告內容應包括下列各項：

一、參加會議經過

本次大會總共接受 33 篇論文，分為 11 個 Sessions 發表，包括 Digital Signatures, Network Security, Wireless Network Security, Security Applications, Cryptanalysis, Multimedia Security, Public Key Encryption, Cryptographic Protocols, Applied Cryptography, System Security, 以及 Algorithms and Evaluation。大會安排兩場 Keynote 演講，由 Jianying Zhou 教授主講的 ``Beyond Basic Password Authentication in Web Applications”，演講內容包含目前密碼身分驗證機制的優缺點分析，並且提供一個同時利用密碼，智慧卡，以及生物特徵進行身分驗證的方法，提供更安全保證的網路應用；以及由K.P.Chow教授主講的 ``Computer Security and Forensics: Defense vs. Post-mortem”，演講內容提到不少關於數位鑑識的相關特性，包括數位證據可複製性，容易修改，以及保存不易，並且以生活上的事例說明介紹數位鑑識的收集分析呈現的流程。

二、　與會心得

此行最主要目的為發表論文 ``Delegable Provable Data Possession for Remote Data in the Clouds”，在私人的遠端資完整性驗證以及公開的遠端資料完整性驗證之間，我們提出了一個平衡點，可授權的遠端資料完整性驗證提供使用者選擇授權哪些人可以進行資料完整性驗證，更適合機密資料的應用環境。

另節錄大會中的幾篇重要論文：

1. Lightweight RFID Mutual Authentication Protocol against Feasible Problems
   這篇論文利用 Shamir 所提出的 SQUASH 方法，減少計算平方和模所需要的運算量，使得 Rabin 的加密方法能夠被 RFID 卡片有效率地處理進行，進而達到輕量化的 RFID 雙向驗證方法，利用 Rabin 加密的安全性，能夠抵禦 RFID 系統中的安全問題：輕量化，非同步攻擊，卡片追蹤，Forward Security 等。

2. Ideal Secret Sharing Schemes with Share Selectability
   此篇論文從 Shamir Secret Sharing 為基礎，從 Secret 更新時 Share 需要重新發布的問題著手，引進了 Share Selectability 的概念，share 與 secret 無關，可以是任意的值，而利用額外的公開資訊來連結 share 與 secret，使得 secret 更新時，share 不用重新發布，更新額外的公開資訊即可。

3. Non-interactive Opening for Ciphertexts Encrypted by Shared Keys
   在 CT-RSA 2008 中，Damg˚ard，Hofheinz，Kiltz，以及 Thorbek 發表了 public key encryption with non-interactive opening (PKENO)，解密者可以向他人產生證明所解得原文的正確性，而不需要洩漏他的解密金鑰。在本篇論文中，利用 Verifiable Random Function 達到了 Secret Key encryption with non-interactive opening (SKENO)，使得 Non interactive opening 不管在 public key 或是 secret key 的環境下都可以發揮作用。

三、考察參觀活動(無是項活動者省略)

無

四、建議

與其他國家相比，國內資安研究與交流活動可以再熱絡，避免閉門造車的現象，英文口語能力也需要加強，增進社交能力。

五、攜回資料名稱及內容

資料名稱

ICICS 2011 國際研討會論文集，Lecture Notes in Computer Science7043 Springer 2011, ISBN 978-3-642-25242-6

六、其他

本次來自台灣的投稿文章只有我們一篇，也順利地獲得採用，本會議雖是中國大陸主辦，但來自日本歐美的文章數量也不少，希望台灣的文章數量和質量上都持續進步，持續培養更多的資安專業人才，提升國際方面的能見度。

Delegable Provable Data Possession for Remote Data in the Clouds

Shiuan-Tzuo Shen and Wen-Guey Tzeng[*]

Department of Computer Science,
National Chiao Tung University,
Hsinchu, Taiwan 30010
{vink,wgtzeng}@cs.nctu.edu.tw

Abstract. Many storage systems need to do authorized verification for data integrity. For example, a user stores his data into cloud storage servers and shares his data with his friends. They check data integrity periodically to ensure data intact. However, they don't want a stranger to check data integrity on their data. Therefore, public verification is undesired in this situation. The user can share his private key to his friends for private verification. However, his friends may reveal his private key to others. In this paper, we proposed the delegable provable data possession (delegable PDP) model to solve this problem. Delegable PDP allows a user to control who can check data integrity of his data, and guarantee that delegated verifiers cannot re-delegate this verification capability to others. Delegable PDP enjoys advantage of authorized verification and convenience of public verification.

We define a delegable PDP model and provide a construction for it. User $\mathcal{U}$ generates verifiable tags of his data and the delegation key $dk_{\mathcal{U}\to\mathcal{V}}$ for delegated verifier $\mathcal{V}$. $\mathcal{U}$ uploads his data, tags, and $dk_{\mathcal{U}\to\mathcal{V}}$ to storage servers. When integrity check, storage servers can use $dk_{\mathcal{U}\to\mathcal{V}}$ to transform $\mathcal{U}$'s tags into the form that $\mathcal{V}$ can verify with his private key $sk_{\mathcal{V}}$. Our model allows $\mathcal{U}$ to revoke $\mathcal{V}$'s verification capability by removing $dk_{\mathcal{U}\to\mathcal{V}}$ from storage servers directly. We prove our protocol secure in the random oracle model. Our protocol achieves proof unforgeability, proof indistinguishability, and delegation key unforgeability.

1 Introduction

Cloud computing provides computing services via networks such that a user can access these services anywhere at any time. For example, Amazon Elastic Compute Cloud (Amazon EC2) provides cloud computation and Amazon Simple Storage Service (Amazon S3) provides cloud storage. Storing data in a cloud storage system is quite convenient. One can share data to other users or synchronize copies in local devices. However, it brings security issues, privacy and integrity, on stored data. Users don't want their data leaked or modified without

[*] This research is supported by parts of NSC projects NSC100-2218-E-009-003-, NSC100-2218-E-009-006-, and NSC100-2219-E-009-005-, Taiwan.

S. Qing et al. (Eds.): ICICS 2011, LNCS 7043, pp. 93–111, 2011.

their permission. In general, encryption can provide data privacy and signature can provide data integrity. Users can encrypt their data and sign ciphertexts before uploading them to cloud storage servers. One way to make sure that a ciphertext is stored intactly is to retrieve the ciphertext together with its signature and verify it. This approach needs large bandwidth since data are retrieved back through networks. Thus, many researchers proposed methods to reduce the bandwidth need.

Ateniese et al. proposed a provable data possession (PDP) model [1]. Their PDP model allows a storage server to generate a probabilistic proof of size $O(1)$ for data integrity check so that a verifier can validate the proof efficiently. Their PDP protocol is asymmetric-key based such that public verification is done by everyone using the public key of the owner. However, public verification is undesirable in many circumstances. In contrast, private verification allows only the owner who possesses the secret key to verify data integrity. The owner can share this secret key to another user for data integrity check. However, the other one may leak this secret key.

In this paper, we define a model for delegable provable data possession (delegable PDP) that allows delegable (authorized) verification. In delegable PDP, a user who owns data can authorize another user to verify data integrity of his data. The authorized user cannot re-delegate this verification capability to others unless the authorized user reveals his private key. The delegable PDP model provides a balance between totally public and totally private integrity checking. Delegable PDP has two goals:

- Proof of data possession. A storage server can generate a valid proof if and only if it really stores the data. This proof can be verified without retrieving back the data from the storage server.
- Delegation of verification capability. A user can delegate his verification capability on his data to another user. The delegated user cannot re-delegate this verification capability to others. The delegated user can verify data integrity with storage servers on behalf of the user. The user can revoke the right of integrity checking from the delegated user directly.

Our delegable PDP model is efficient. To delegate, data owner $\mathcal{U}$ doesn't need to re-tag his data for delegated verifier $\mathcal{V}$. Instead, $\mathcal{U}$ generates the delegation key $dk_{\mathcal{U}\rightarrow\mathcal{V}}$ and uploads it to storage servers. Thus, $\mathcal{V}$ doesn't store and doesn't know $dk_{\mathcal{U}\rightarrow\mathcal{V}}$. To revoke, $\mathcal{U}$ sends the revoking command of deleting $dk_{\mathcal{U}\rightarrow\mathcal{V}}$ to storage servers directly. The cost of delegation is lightweight.

1.1 Delegable Provable Data Possession

There are three roles, user (data owner) $\mathcal{U}$, delegated verifier $\mathcal{V}$, and storage server $\mathcal{S}$, in the delegable PDP model. The data are stored in $\mathcal{S}$ after tagged by $\mathcal{U}$'s private key $sk_{\mathcal{U}}$. For delegation, $\mathcal{U}$ computes the delegation key $dk_{\mathcal{U}\rightarrow\mathcal{V}}$ by using $sk_{\mathcal{U}}$ and $\mathcal{V}$'s public key $pk_{\mathcal{V}}$, and sends it to $\mathcal{S}$. $\mathcal{S}$ transforms the tags of the data by using $dk_{\mathcal{U}\rightarrow\mathcal{V}}$ such that $\mathcal{V}$ can use his private key $sk_{\mathcal{V}}$ to verify the data by the transformed tags.

A delegable PDP scheme has three phases: the setup phase, the delegation phase, and the integrity check phase. The setup phase consists of three algorithms, Setup, KeyGen, and TagGen, as follows:

- Setup$(1^k) \rightarrow \pi$. It is a probabilistic polynomial time algorithm run by the system manager to set up a delegable PDP system. Setup takes as input the security parameter k and outputs the public parameter π.
- KeyGen$(\pi) \rightarrow (sk, pk)$. It is a probabilistic polynomial time algorithm run by a user to generate his key pair. KeyGen takes as input the public parameter π and outputs a private-public key pair (sk, pk) for the user.
- TagGen$(\pi, sk, m) \rightarrow (\sigma, t)$. It is a deterministic polynomial time algorithm run by a user to generate verifiable tags for his data. TagGen takes as input the public parameter π, the user's private key sk, and the user's data m, and outputs a tag σ for m and an identifier t for σ.

The delegation phase consists of two algorithms, GenDK and VrfyDK, as follows:

- GenDK$(\pi, sk_{\mathcal{U}}, pk_{\mathcal{V}}) \rightarrow dk_{\mathcal{U}\rightarrow\mathcal{V}}$. It is a deterministic polynomial time algorithm run by $\mathcal{U}$ to generate a delegation key for $\mathcal{V}$. GenDK takes as input the public parameter π, $\mathcal{U}$'s private key $sk_{\mathcal{U}}$, and $\mathcal{V}$'s public key $pk_{\mathcal{V}}$, and outputs the delegation key $dk_{\mathcal{U}\rightarrow\mathcal{V}}$.
- VrfyDK$(\pi, dk_{\mathcal{U}\rightarrow\mathcal{V}}, pk_{\mathcal{U}}, pk_{\mathcal{V}}) \rightarrow \{true, false\}$. It is a deterministic polynomial time algorithm run by $\mathcal{S}$ to verify delegation keys. VrfyDK takes as input the public parameter π, the delegation key $dk_{\mathcal{U}\rightarrow\mathcal{V}}$, $\mathcal{U}$'s public key $pk_{\mathcal{U}}$, and $\mathcal{V}$'s public key $pk_{\mathcal{V}}$, and outputs the verification result.

The integrity check phase consists of three algorithms, GenChal, GenProof, and VrfyProof, as follows:

- GenChal$(\pi, t) \rightarrow chal$. It is a probabilistic polynomial time algorithm run by $\mathcal{V}$ to generate a challenge to $\mathcal{S}$ for $\mathcal{U}$'s stored data. GenChal takes as input the public parameter π and tag identifier t, and outputs the challenge $chal$.
- GenProof$(\pi, m, \sigma, dk_{\mathcal{U}\rightarrow\mathcal{V}}, chal) \rightarrow pf_{chal,\mathcal{V}}$. It is a probabilistic polynomial time algorithm run by $\mathcal{S}$ to generate a proof for integrity of the challenged data. GenProof takes as input the public parameter π, the stored data m, the tag σ for m, the delegation key $dk_{\mathcal{U}\rightarrow\mathcal{V}}$, and the challenge $chal$, and outputs the proof $pf_{chal,\mathcal{V}}$.
- VrfyProof$(\pi, chal, pf_{chal,\mathcal{V}}, t, sk_{\mathcal{V}}) \rightarrow \{true, false\}$. It is a deterministic polynomial time algorithm run by $\mathcal{V}$ to verify a proof from $\mathcal{S}$. VrfyProof takes as input the public parameter π, the challenge $chal$, the proof $pf_{chal,\mathcal{V}}$, the identifier t, and $\mathcal{V}$'s private key $sk_{\mathcal{V}}$, and outputs the verification result.

1.2 Related Work

Ateniese et al. [1] defined the PDP model. They proposed an asymmetric-key based PDP construction which uses homomorphic verifiable tags on stored data. Under the homomorphic property, storage servers can generate proofs for any linear combination of the stored data. Later on, Ateniese et al. [2] proposed a symmetric-key PDP construction which supports dynamic operations on stored data. Their construction is scalable and efficient. However, the number of data possession checkings is limited by the number of embedded tokens. Erway et al. [12] proposed dynamic provable data possession (DPDP) which uses rank-based skip list to support dynamic data operations. Ateniese et al. [3] proposed a framework for constructing public-key based PDP protocols. Their framework builds public-key homomorphic linear authenticators (HLAs) from public-key identification schemes, which satisfy certain homomorphic properties, and uses the HLA as a building block to construct PDP protocols.

Juels and Kaliski [14] proposed the proofs of retrievability (POR) model. POR ensures that stored data can be retrieved by users, while PDP ensures that data are stored in storage servers. Juels and Kaliski's construction embeds sentinels (verifying information of precomputed challenge-response pairs) into stored data, and the number of checkings is limited by the number of embedded sentinels. Later on, Shacham and Waters [15] proposed a compact POR which achieves an unlimited number of checkings. Bowers et al. [7] proposed a theoretical framework of designing POR protocols. This framework employs two layers of error correcting codes which recover user data from a series of responses. Their framework improves previous results of POR and has security proved in the fully Byzantine adversarial model. Wang et al. [17] proposed a POR scheme which supports dynamic operations and public verification on stored data. Their construction uses Merkle hash tree to support data dynamics.

To simultaneously achieve high availability and integrity checking for stored data, multiple replicas or the coding theory can be employed. Curtmola et al. [10] proposed MR-PDP that makes sure each unique replica exists in storage servers. Curtmola et al. [9] proposed a robust remote data integrity checking method that uses forward error correction codes. Later on, Bowers et al. [6] proposed HAIL which provides a high-availability and integrity layer for cloud storage. HAIL uses erasure codes on the single server layer and multiple sever layer respectively. It ensures data retrievability among distributed storage servers.

A cloud storage system may be viewed as a set of distributed storage servers. One can use the network coding technique to dispatch data to storage servers. For this model, Chen et al. [8] proposed a remote data integrity checking method for network coding-based distributed storage systems.

Wang et al. [16] proposed privacy-preserving public auditing for data storage security in cloud computing. Public data integrity checking may leak information about stored data by proofs to verifiers. Wang et al. use a blinding technique to hide information about stored data in proofs.

2 Preliminary

Our delegable PDP protocol uses the *bilinear map*. The security of our protocol is based on the *truncated (decision) bilinear Diffie-Hellman exponent assumption*, the *inverse computation Diffie-Hellman assumption*, and the *knowledge of exponent assumption* in the random oracle model.

Bilinear Map. Let q be a large prime, $G = \langle g \rangle$ and $G_T = \langle g_T \rangle$ be two multiplicative groups of prime order q. A bilinear map $\hat{e} : G \times G \rightarrow G_T$ should satisfy the following properties:

- Bilinearity. $\forall x, y \in \mathbb{Z}_q$, $\hat{e}(g^x, g^y) = \hat{e}(g, g)^{xy}$.
- Non-Degeneration. $\hat{e}(g, g) = g_T$.
- Computability. $\forall x, y \in \mathbb{Z}_q$, $\hat{e}(g^x, g^y)$ can be computed in polynomial time.

Truncated Bilinear Diffie-Hellman Exponent Assumption. Boneh et al. introduced the *bilinear Diffie-Hellman exponent* (BDHE) problem [4,5]. Later on, Gentry introduced two variants: the *augmented bilinear Diffie-Hellman exponent* (ABDHE) problem and the *truncated version* of the ABDHE problem [13].

The ℓ-BDHE problem is that: given a vector

$$\left(g', g, g^{\alpha}, g^{\alpha^2}, \ldots, g^{\alpha^{\ell}}, g^{\alpha^{\ell+2}}, g^{\alpha^{\ell+3}}, \ldots, g^{\alpha^{2\ell}}\right) \in G^{2\ell+1} \ ,$$

output $\hat{e}(g, g')^{\alpha^{\ell+1}} \in G_T$. The *truncated version* of the ℓ-BDHE problem, omitting ($g^{\alpha^{\ell+2}}$, $g^{\alpha^{\ell+3}}$, ..., $g^{\alpha^{2\ell}}$) from the input vector, is defined as that: given a vector

$$\left(g', g, g^{\alpha}, g^{\alpha^2}, \ldots, g^{\alpha^{\ell}}\right) \in G^{\ell+2} \ ,$$

output $\hat{e}(g, g')^{\alpha^{\ell+1}} \in G_T$. The advantage for an algorithm $\mathcal{A}$ that solves the *truncated* ℓ-BDHE problem is defined as:

$$\Pr\left[\mathcal{A}(g', g, g^{\alpha}, g^{\alpha^2}, \ldots, g^{\alpha^{\ell}}) = \hat{e}(g, g')^{\alpha^{\ell+1}} : g, g' \in_R G, \alpha \in_R \mathbb{Z}_q\right] \ .$$

The advantage for an algorithm $\mathcal{A}$ that solves the *truncated decisional* ℓ-BDHE problem is defined as:

$$\begin{aligned}\Big|&\Pr[\mathcal{A}(g', g, g^{\alpha}, g^{\alpha^2}, \ldots, g^{\alpha^{\ell}}, \hat{e}(g, g')^{\alpha^{\ell+1}}) = 0 : g, g' \in_R G, \alpha \in_R \mathbb{Z}_q] \ - \\ &\Pr[\mathcal{A}(g', g, g^{\alpha}, g^{\alpha^2}, \ldots, g^{\alpha^{\ell}}, Z) = 0 : g, g' \in_R G, \alpha \in_R \mathbb{Z}_q, Z \in_R G_T]\Big| \ .\end{aligned}$$

Definition 1. *We say that the truncated (decisional) BDHE assumption is (t, ϵ, ℓ)-secure if no t-time algorithms have advantage over ϵ in solving the truncated (decisional) ℓ-BDHE problem.*

Inverse Computational Diffie-Hellman Assumption. The InvCDH problem is defined as that: given $(g, g^{\alpha}) \in G^2$ as input, output $g^{\frac{1}{\alpha}} \in G$. The advantage for a probabilistic algorithm $\mathcal{A}$ to solve the InvCDH problem is:

$$\Pr\left[\mathcal{A}(g, g^{\alpha}) = g^{\frac{1}{\alpha}} : \alpha \in_R \mathbb{Z}_q\right] \ .$$

Definition 2. *We say that the InvCDH assumption is (t, ϵ)-secure if no t-time algorithms have advantage over ϵ in solving the InvCDH problem*

Knowledge of Exponent Assumption. Damgard introduced the *knowledge of exponent assumption* ($\mathsf{KEA1}$) [11]. Consider the problem: given $(g, g^{\alpha}) \in G^2$, output $(C, Y) \in G^2$ such that $C^{\alpha} = Y$. One way to output the pair is to choose $c \in_R \mathbb{Z}_q$ and let $(C, Y) = (g^c, g^{\alpha c})$. The $\mathsf{KEA1}$ says that this is the only way to output such a pair in polynomial time. That is, if an adversary $\mathcal{A}$ takes (g, g^{α}) as input and outputs (C, Y) such that $C^{\alpha} = Y$, he must know the exponent c of $g^c = C$. There exists an extractor $\overline{\mathcal{A}}$ who extracts the exponent c such that $g^c = C$ when he is given the same inputs as $\mathcal{A}$'s.

3 Construction

In this section, we provide a delegable PDP scheme. Let k be the security parameter, q be a large prime with $|q| = k$, and $G = \langle g \rangle$ and $G_T = \langle g_T \rangle$ be two order-q multiplicative groups with a bilinear map $\hat{e} : G \times G \rightarrow G_T$. The system manager chooses three cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow G$, $H_2 : G \rightarrow G$, and $H_3 : (\mathbb{Z}_q)^* \rightarrow G$. The public parameter is $\pi = (q, G, g, G_T, g_T, \hat{e}, H_1, H_2, H_3)$.

Key Generation. $\mathcal{U}$ chooses $x \in_R \mathbb{Z}_q$ as his private key $sk_{\mathcal{U}}$ and computes g^x as his public key $pk_{\mathcal{U}}$ and $H_2(g^x)^x$ as his key token $kt_{\mathcal{U}}$. $\mathcal{U}$'s key tuple is $(sk_{\mathcal{U}}, pk_{\mathcal{U}}, kt_{\mathcal{U}}) = (x, g^x, H_2(g^x)^x)$. $\mathcal{U}$ registers $pk_{\mathcal{U}}$ to the system manager.

Tag Computation. $\mathcal{U}$ has data $\mathcal{M} = (m_1, m_2, \ldots, m_n)$, each block k-bit long, and would like to store them in $\mathcal{S}$. $\mathcal{U}$ chooses data identifier $h_{\mathcal{M}} \in_R G$ and tag identifier seed $T_{\mathcal{M}} \in_R \{0,1\}^*$ for $\mathcal{M}$. $\mathcal{U}$ may have many different data. Thus, he needs to choose a unique data identifier for each of his data. Each block m_i is tagged to a homomorphic verifiable tag σ_i which is identified by $h_{\mathcal{M}}$ and tag identifier $T_{\mathcal{M}}||i$. $\mathcal{U}$ computes these homomorphic verifiable tags $\Sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ for $\mathcal{M}$ as follows:

$$\sigma_i = [H_1(T_{\mathcal{M}}||i) h_{\mathcal{M}}^{m_i}]^{sk_{\mathcal{U}}} \quad , \text{for } 1 \leq i \leq n \ .$$

$\mathcal{U}$ uploads $(\mathcal{M}, h_{\mathcal{M}}, \Sigma)$ to $\mathcal{S}$, and holds $(h_{\mathcal{M}}, T_{\mathcal{M}})$ for identifying and verifying Σ.

Delegation. $\mathcal{V}$ gives his key token $kt_{\mathcal{V}}$ to $\mathcal{U}$ over a secure channel, and obtains $h_{\mathcal{M}}$ and $T_{\mathcal{M}}$ from $\mathcal{U}$. $\mathcal{U}$ uses $\mathcal{V}$'s public key $pk_{\mathcal{V}}$ to verify validity of $kt_{\mathcal{V}}$ by checking whether $\hat{e}(g, kt_{\mathcal{V}}) = \hat{e}(pk_{\mathcal{V}}, H_2(pk_{\mathcal{V}}))$. Then, $\mathcal{U}$ computes the delegation key

$$dk_{\mathcal{U} \rightarrow \mathcal{V}} = kt_{\mathcal{V}}^{1/sk_{\mathcal{U}}}$$

and gives it to $\mathcal{S}$. $\mathcal{S}$ uses $pk_{\mathcal{U}}$ and $pk_{\mathcal{V}}$ to verify validity of $dk_{\mathcal{U} \rightarrow \mathcal{V}}$ by checking whether $\hat{e}(pk_{\mathcal{U}}, dk_{\mathcal{U} \rightarrow \mathcal{V}}) = \hat{e}(pk_{\mathcal{V}}, H_2(pk_{\mathcal{V}}))$. To revoke $\mathcal{V}$, $\mathcal{U}$ commands $\mathcal{S}$ to remove $dk_{\mathcal{U} \rightarrow \mathcal{V}}$ from its storage directly.

Integrity Check. To check integrity of $\mathcal{M}$, $\mathcal{V}$ chooses coefficients $C = (c_1, c_2, \ldots, c_n) \in_R \mathbb{Z}_q^n$ and gives $\mathcal{S}$ the challenge

$$chal = (C,\ C',\ C'') = (C,\ h_{\mathcal{M}}^s,\ H_3(C)^s)\,, \quad \text{where } s \in_R \mathbb{Z}_q\ .$$

After receiving *chal*, $\mathcal{S}$ verifies it by checking whether $\hat{e}(C', H_3(C)) = \hat{e}(h_{\mathcal{M}}, C'')$. If so, $\mathcal{S}$ uses $(\mathcal{M}, \Sigma, dk_{\mathcal{U}\to\mathcal{V}}, chal)$ to generate a proof $pf_{chal,\mathcal{V}} = (\rho, V, V', V'', V''')$ and gives it to $\mathcal{V}$ as a response. $pf_{chal,\mathcal{V}}$ is computed as follows:

$$pf_{chal,\mathcal{V}} = \left(\hat{e}(\prod_{i=1}^{n} \sigma_i^{c_i},\ dk_{\mathcal{U}\to\mathcal{V}})^t,\ C'^{\sum_{i=1}^n c_i m_i},\ C''^{\sum_{i=1}^n c_i m_i}, H_2(pk_{\mathcal{V}})^t,\ g^t\right), \text{ where } t \in_R \mathbb{Z}_q.$$

After receiving $pf_{chal,\mathcal{V}}$, $\mathcal{V}$ uses $(sk_{\mathcal{V}}, h_{\mathcal{M}}, T_{\mathcal{M}}, C, s)$ to verify $pf_{chal,\mathcal{V}}$ by checking whether

- $\rho^s = \hat{e}\left(\prod_{i=1}^n H_1(T_{\mathcal{M}}||i)^{sc_i} V,\ V''\right)^{sk_{\mathcal{V}}}$
- $\hat{e}\,(V,\ H_3(C)) = \hat{e}\,(h_{\mathcal{M}},\ V')$
- $\hat{e}\,(V'',\ g) = \hat{e}\,(H_2(pk_{\mathcal{V}}),\ V''')$

$\mathcal{V}$ can verify data integrity of $\mathcal{M}$ multiple times to achieve a desire security level.

Correctness. Although tag $\sigma = [H_1(T_{\mathcal{M}})h_{\mathcal{M}}^m]^{sk_{\mathcal{U}}}$ is called homomorphic verifiable in the literature, it is really not homomorphic. Instead, σ is combinably verifiable since we can combine multiple tags together and verify them at the same time. Nevertheless, we cannot obtain a tag for the combined data. For example, combing $\sigma_i = [H_1(T_{\mathcal{M}}||i)h_{\mathcal{M}}^{m_i}]^{sk_{\mathcal{U}}}$ and $\sigma_j = \left[H_1(T_{\mathcal{M}}||j)h_{\mathcal{M}}^{m_j}\right]^{sk_{\mathcal{U}}}$ together results in $\sigma' = \left[H_1(T_{\mathcal{M}}||i)H_1(T_{\mathcal{M}}||j)h_{\mathcal{M}}^{m_i+m_j}\right]^{sk_{\mathcal{U}}}$. Although we have $m_i + m_j$ in the exponent of $h_{\mathcal{M}}$, we don't have $H_1(T_{\mathcal{M}}||k)$ in the combined tag σ' for some k (treat $m_i + m_j = m_k$). The tag is unforgeable, proved in Sect. 4.1 (proof unforgeability implies tag unforgeability). It is hard to obtain $\sigma' = \left[H_1(T_{\mathcal{M}}||k)h_{\mathcal{M}}^{m_i+m_j}\right]^{sk_{\mathcal{U}}}$ without the knowledge of private key $sk_{\mathcal{U}}$.

In integrity check, $\mathcal{V}$ chooses coefficients $C = (c_1, c_2, \ldots, c_n)$ and then $\mathcal{S}$ combines stored tags $\Sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ as $\prod_{i=1}^n \sigma_i^{c_i} = \left[\prod_{i=1}^n H_1(T_{\mathcal{M}}||i)^{c_i} \times h_{\mathcal{M}}^{\sum_{i=1}^n c_i m_i}\right]^{sk_{\mathcal{U}}}$ by C. If $\mathcal{S}$ deviates, the combination will not be identical to $\prod_{i=1}^n H_1(T_{\mathcal{M}}||i)^{c_i}$. Once $\mathcal{S}$ combines these tags correctly, we have $\sum_{i=1}^n c_i m_i$ in the exponent of $h_{\mathcal{M}}$. On the other hand, $\mathcal{S}$ has to use stored data $\mathcal{M} = (m_1, m_2, \ldots, m_n)$ to compute $V = C'^{\sum_{i=1}^n c_i m_i} = (h_{\mathcal{M}}^s)^{\sum_{i=1}^n c_i m_i}$ and $V' = C''^{\sum_{i=1}^n c_i m_i} = [H_3(C)^s]^{\sum_{i=1}^n c_i m_i}$. Our verification, $\rho^s = \hat{e}\left(\prod_{i=1}^n H_1(T_{\mathcal{M}}||i)^{sc_i} V, V''\right)^{sk_{\mathcal{V}}}$, checks whether ρ contains the correct combination $\prod_{i=1}^n H_1(T_{\mathcal{M}}||i)^{c_i}$ and whether V contains the same exponent

$\sum_{i=1}^{n} c_i m_i$, with respect to $h_{\mathcal{M}}^{s}$, as that in ρ, with respect to $h_{\mathcal{M}}$. If $\mathcal{S}$ passes this verification, he possesses $\mathcal{M}$.

Let's examine the verification equations. Assume that $pf_{chal,\mathcal{V}}$ is well-formed and $chal = (C, C', C'') = (C, h_{\mathcal{M}}^{s}, H_3(C)^s)$, that is,

$$\begin{aligned}
\rho &= \hat{e}(\prod_{i=1}^{n} \sigma_i^{c_i}, dk_{\mathcal{U}\to\mathcal{V}})^t && (1)\\
V &= C'^{\sum_{i=1}^{n} c_i m_i} = h_{\mathcal{M}}^{s\sum_{i=1}^{n} c_i m_i} && (2)\\
V' &= C''^{\sum_{i=1}^{n} c_i m_i} = H_3(C)^{s\sum_{i=1}^{n} c_i m_i} && (3)\\
V'' &= H_2(pk_{\mathcal{V}})^t && (4)\\
V''' &= g^t && (5)
\end{aligned}$$

We have:

- $\rho^s = \hat{e}(\prod_{i=1}^{n} H_1(T_{\mathcal{M}}||i)^{sc_i} V, V'')^{sk_{\mathcal{V}}}$ by (1), (2), and (4)

$$\begin{aligned}
\rho^s &= \hat{e}(\prod_{i=1}^{n} \sigma_i^{c_i}, dk_{\mathcal{U}\to\mathcal{V}})^{ts}\\
&= \hat{e}(\prod_{i=1}^{n} (H_1(T_{\mathcal{M}}||i) h_{\mathcal{M}}^{m_i})^{sk_{\mathcal{U}} c_i}, H_2(pk_{\mathcal{V}})^{sk_{\mathcal{V}}/sk_{\mathcal{U}}})^{ts}\\
&= \hat{e}(\prod_{i=1}^{n} (H_1(T_{\mathcal{M}}||i) h_{\mathcal{M}}^{m_i})^{sc_i}, H_2(pk_{\mathcal{V}})^{sk_{\mathcal{V}}})^{t}\\
&= \hat{e}(\prod_{i=1}^{n} H_1(T_{\mathcal{M}}||i)^{sc_i} h_{\mathcal{M}}^{s\sum_{i=1}^{n} c_i m_i}, H_2(pk_{\mathcal{V}})^{t})^{sk_{\mathcal{V}}}\\
&= \hat{e}(\prod_{i=1}^{n} H_1(T_{\mathcal{M}}||i)^{sc_i} V, V'')^{sk_{\mathcal{V}}}
\end{aligned}$$

- $\hat{e}(V, H_3(C)) = \hat{e}(h_{\mathcal{M}}, V')$ by (2) and (3)

$$\begin{aligned}
\hat{e}(V, H_3(C)) &= \hat{e}(h_{\mathcal{M}}^{s\sum_{i=1}^{n} c_i m_i}, H_3(C))\\
&= \hat{e}(h_{\mathcal{M}}, H_3(C)^{s\sum_{i=1}^{n} c_i m_i})\\
&= \hat{e}(h_{\mathcal{M}}, V')
\end{aligned}$$

- $\hat{e}(V'', g) = \hat{e}(H_2(pk_{\mathcal{V}}), V''')$ by (4) and (5)

$$\begin{aligned}
\hat{e}(V'', g) &= \hat{e}(H_2(pk_{\mathcal{V}})^t, g)\\
&= \hat{e}(H_2(pk_{\mathcal{V}}), g^t)\\
&= \hat{e}(H_2(pk_{\mathcal{V}}), V''')
\end{aligned}$$

3.1 Performance

We analyze performance of our construction in three aspects: the computation cost of each algorithm, the storage cost of each party, and the communication cost of each phase. Table 1 shows the computation cost of each algorithm.[1] We measure the numbers of additions in $\mathbb{Z}_q$, multiplications in G, scalar exponentiations in G, hashes, and pairings.

Table 1. Computation cost of each algorithm

Algorithm	Addition	Multiplication	Scalar Exponentiation	Hash	Pairing
Setup	0	0	0	0	0
KeyGen	0	0	2	1	0
TagGen	0	n	$2n$	n	0
GenDK	0	0	1	1	2
VrfyDK	0	0	0	1	2
GenChal	0	0	2	1	0
GenProof	$n-1$	$2n-1$	$n+5$	2	3
VrfyProof	0	n	$n+3$	$n+2$	5

– n is the number of data blocks.

Table 2 shows the storage cost of each party. User $\mathcal{U}$ stores his key tuple $(sk_{\mathcal{U}}, pk_{\mathcal{U}}, kt_{\mathcal{U}})$, data identifier $h_{\mathcal{M}}$, and tag identifier seed $T_{\mathcal{M}}$. Delegated verifier $\mathcal{V}$ stores his key tuple $(sk_{\mathcal{V}}, pk_{\mathcal{V}}, kt_{\mathcal{V}})$, data identifier $h_{\mathcal{M}}$, and tag identifier seed $T_{\mathcal{M}}$. Storage server $\mathcal{S}$ stores $\mathcal{U}$'s data $\mathcal{M}$, data identifier $h_{\mathcal{M}}$, tags Σ, and the delegation keys. Table 3 shows the communication cost of each phase. In setup phase, $\mathcal{U}$ uploads $\mathcal{M}$, Σ, and $h_{\mathcal{M}}$ to $\mathcal{S}$. In delegation phase, $\mathcal{V}$ gives $\mathcal{U}$ his key token $kt_{\mathcal{V}}$. $\mathcal{U}$ gives $h_{\mathcal{M}}$ and $T_{\mathcal{M}}$ to $\mathcal{V}$, and gives the delegation key $dk_{\mathcal{U}\to\mathcal{V}}$ to $\mathcal{S}$. In integrity check phase, $\mathcal{V}$ gives $\mathcal{S}$ the challenge $chal$,[2] and $\mathcal{S}$ gives $\mathcal{V}$ the proof $pf_{chal,\mathcal{V}}$.

[1] To achieve better performance, one can choose binary coefficient $c_i \in \{0,1\}$, $1 \leq i \leq n$, to reduce computation on multiplications and scalar exponentiations. Thus, in algorithm GenProof, we don't need to do scalar exponentiations on σ_i to compute $\sigma_i^{c_i}$, and multiplications on m_i to compute $c_i m_i$. Thus, it reduces the computation cost from $2n-1$ multiplications in G and $n+5$ scalar exponentiations in G to $n-1$ multiplications in G and 5 scalar exponentiations in G for GenProof. And in algorithm VrfyProof, we don't really do scalar exponentiations on $H_1(T_{\mathcal{M}}||i)$ to compute $H_1(T_{\mathcal{M}}||i)^{c_i}$, either. Thus, it reduces the computation cost from $n+3$ scalar exponentiations in G to 3 scalar exponentiations in G for VrfyProof.

[2] To reduce the communication cost on transmitting $chal$, one can choose a random seed c of size ℓ' for computing coefficients $c_i = H(c,i)$, $1 \leq i \leq n$, and send c only. Thus, it reduces the communication cost from $nk + 6p + p_T$ bits to $\ell' + 6p + p_T$ bits in integrity check phase.

Table 2. Storage cost of each party

Party	Storage Cost (Bit)
User	$k + 3p + \ell$
Delegated Verifier	$k + 3p + \ell$
Storage Server	$nk + (1 + n + v)p$

¯ k is the security parameter
¯ p is the size of an element in G
¯ l is the length of tag identifier seed $T_{\mathcal{M}}$
¯ n is the number of data blocks
¯ v is the number of delegated verifiers

Table 3. Communication cost of each phase

Phase	Communication Cost (Bit)
Setup	$\mathcal{U} \rightarrow \mathcal{S} : nk + p + np$
Delegation	$\mathcal{V} \leftrightarrow \mathcal{U} : 2p + \ell$ $\mathcal{U} \rightarrow \mathcal{S} : p$
Integrity Check	$\mathcal{V} \leftrightarrow \mathcal{S} : nk + 6p + p_T$

¯ k is the security parameter
¯ p is the size of an element in G
¯ p_T is the size of an element in G_T
¯ l is the length of tag identifier $h_{\mathcal{M}}$
¯ n is the number of data blocks

4 Security Analysis

The security requirements of a delegable PDP model consists of **proof unforgeability**, **proof indistinguishability**, and **delegation key unforgeability**. We introduce the security games in the rest subsections and prove that our construction satisfies these security requirements in the random oracle model.

4.1 Proof Unforgeability

This game models the notion that a storage server cannot modify stored data without being detected by verifiers. In this game, the challenger $\mathcal{C}$ plays the role of the verifier and the adversary $\mathcal{A}$ plays the role of the storage server. $\mathcal{A}$ is given the access right to oracles $\mathcal{O}_{\mathsf{Tag}}$ and $\mathcal{O}_{\mathsf{DK}}$. $\mathcal{A}$ chooses data adaptively and obtains corresponding tags. Once $\mathcal{A}$ decides the target data $\mathcal{M}^*$, he modifies $\mathcal{M}^*$ to $\mathcal{M}'$ such that $\mathcal{M}' \neq \mathcal{M}^*$ and receives a challenge from $\mathcal{C}$. If $\mathcal{A}$ returns a proof that passes the verification algorithm, he wins this game.

The proof unforgeability game $\mathsf{Game}^{\mathsf{PF-UF}}$ is as follows:

Setup. $\mathcal{C}$ generates public parameter π, user $\mathcal{U}$'s key tuple $(sk_{\mathcal{U}}, pk_{\mathcal{U}}, kt_{\mathcal{U}})$, delegated verifier $\mathcal{V}$'s key tuple $(sk_{\mathcal{V}}, pk_{\mathcal{V}}, kt_{\mathcal{V}})$, and the delegation key $dk_{\mathcal{U}\rightarrow\mathcal{V}}$. $\mathcal{C}$ forwards $(\pi, pk_{\mathcal{U}}, pk_{\mathcal{V}}, dk_{\mathcal{U}\rightarrow\mathcal{V}})$ to $\mathcal{A}$.

Query. $\mathcal{A}$ queries oracle $\mathcal{O}_{\mathsf{Tag}}$ and oracle $\mathcal{O}_{\mathsf{DK}}$ to obtain tags and delegation keys.

- $\mathcal{O}_{\mathsf{Tag}}$: $\mathcal{A}$ chooses data $\mathcal{M}$ and obtains tags Σ, data identifier $h_{\mathcal{M}}$, and tag identifier seed $T_{\mathcal{M}}$ for $\mathcal{M}$.
- $\mathcal{O}_{\mathsf{DK}}$: $\mathcal{A}$ chooses a user $\mathcal{U}'$ and obtains the delegation key $dk_{\mathcal{U}\to\mathcal{U}'}$.

Challenge. After the query phase, $\mathcal{A}$ indicates which $\mathcal{O}_{\mathsf{Tag}}$-oracle query is the target, denoted as ($\mathcal{M}^*$, Σ^*, $h_{\mathcal{M}^*}$, $T_{\mathcal{M}^*}$), and modifies $\mathcal{M}^* = (m_1^*, m_2^*, \ldots, m_n^*)$ to $\mathcal{M}' = (m_1', m_2', \ldots, m_n')$ such that $\mathcal{M}' \neq \mathcal{M}^*$ $(\exists i, m_i' \neq m_i^*)$. $\mathcal{C}$ gives challenge $chal = (C, h_{\mathcal{M}^*}^s, H_3(C)^s)$.
Answer. $\mathcal{A}$ returns proof $pf_{chal,\mathcal{V}}$ by using $\mathcal{M}'$. $\mathcal{A}$ wins $\mathsf{Game}^{\mathsf{PF-UF}}$ if $\mathsf{VrfyProof}(\pi,\ chal,\ pf_{chal,\mathcal{V}},\ h_{\mathcal{M}^*},\ T_{\mathcal{M}^*},\ sk_{\mathcal{V}}) = true$ and $\mathcal{M}' \neq \mathcal{M}^*$ $(\exists i, m_i' \neq m_i^*)$. The advantage $Adv_{\mathcal{A}}^{\mathsf{Pf-UF}}$ is defined as Pr[$\mathcal{A}$ wins $\mathsf{Game}^{\mathsf{PF-UF}}$].

We show that our scheme is proof unforgeable under the *truncated* 1-BDHE assumption and the KEA1.

Theorem 1. *If the truncated BDHE problem is $(t, \epsilon, 1)$-secure, the above scheme is $(t - q_1t_1 - q_2t_2 - q_Tt_T - q_Kt_K - 2t_{\overline{A}}, \frac{2^\ell}{2^\ell-(q_1+q_T)q_T}\frac{2^k}{2^k-1}\epsilon)$ proof unforgeable in the random oracle model, where hash functions H_1 and H_2 are modeled as random oracles $\mathcal{O}_{\mathsf{H}_1}$ and $\mathcal{O}_{\mathsf{H}_2}$, (q_1, q_2, q_T, q_K) are the numbers of times that an adversary queries $(\mathcal{O}_{\mathsf{H}_1}, \mathcal{O}_{\mathsf{H}_2}, \mathcal{O}_{\mathsf{Tag}}, \mathcal{O}_{\mathsf{DK}})$-oracles, (t_1, t_2, t_T, t_K) are the time used by $(\mathcal{O}_{\mathsf{H}_1}, \mathcal{O}_{\mathsf{H}_2}, \mathcal{O}_{\mathsf{Tag}}, \mathcal{O}_{\mathsf{DK}})$-oracles to respond an oracle query, $t_{\overline{A}}$ is the time used by the KEA1 extractor $\overline{A}$ to extract an exponent, k is the security parameter, and ℓ is the bit-length of a tag identifier seed.*

Proof. Let $\mathcal{A}$ be a probabilistic black-box adversary who wins the proof unforgeability game $\mathsf{Game}^{\mathsf{PF-UF}}$ with advantage ϵ' in time t'. We construct an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to solve the *truncated* 1-BDHE problem as follows:

Setup. Given an instance (g, g^α, g') of the *truncated* 1-BDHE problem, $\mathcal{B}$ sets the public parameter $\pi = (q, G, g, G_T, g_T, \hat{e}, H_3)$, user $\mathcal{U}$'s key tuple $(sk_{\mathcal{U}}, pk_{\mathcal{U}}, kt_{\mathcal{U}}) = (\alpha u, g^{\alpha u}, \mathsf{H}_2(g^{\alpha u})^{\alpha u})$, where $\mathsf{H}_2(g^{\alpha u}) = g^{\alpha u'}$ and $u, u' \in_R \mathbb{Z}_q$, and delegated verifier $\mathcal{V}$'s key tuple $(sk_{\mathcal{V}}, pk_{\mathcal{V}}, kt_{\mathcal{V}}) = (\alpha v, g^{\alpha v}, \mathsf{H}_2(g^{\alpha v})^{\alpha v})$, where $\mathsf{H}_2(g^\alpha v) = g^{\alpha v'}$ and $v, v' \in_R \mathbb{Z}_q$. Then $\mathcal{B}$ computes the delegation key

$$dk_{\mathcal{U}\to\mathcal{V}} = \mathsf{H}_2(pk_{\mathcal{V}})^{sk_{\mathcal{V}}/sk_{\mathcal{U}}} = (g^{\alpha v'})^{\alpha v/\alpha u} = g^{\alpha v v'/u}$$

and invokes $\mathcal{A}$ as a subroutine: $\mathcal{A}^{\mathcal{O}_{\mathsf{H}_1}, \mathcal{O}_{\mathsf{H}_2}, \mathcal{O}_{\mathsf{Tag}}, \mathcal{O}_{\mathsf{DK}}}(\pi, pk_{\mathcal{U}}, pk_{\mathcal{V}}, dk_{\mathcal{U}\to\mathcal{V}})$.

Query. $\mathcal{A}$ can query oracles $\mathcal{O}_{\mathsf{H}_1}$, $\mathcal{O}_{\mathsf{H}_2}$, $\mathcal{O}_{\mathsf{Tag}}$, and $\mathcal{O}_{\mathsf{DK}}$ during his execution. $\mathcal{B}$ handles these oracles as follows:

- $\mathcal{O}_{\mathsf{H}_1}$. $\mathcal{B}$ maintains a table $\mathcal{T}_{\mathsf{H}_1} = \{(x, \mathsf{H}_1(x), r)\}$ to look up the $\mathcal{O}_{\mathsf{H}_1}$-query records. $\mathcal{B}$ takes $x \in \{0,1\}^*$ as input and outputs y if record $(x, y, *)$ exists in $\mathcal{T}_{\mathsf{H}_1}$. Otherwise, $\mathcal{B}$ outputs $\mathsf{H}_1(x) = g^r$ and inserts (x, g^r, r) into $\mathcal{T}_{\mathsf{H}_1}$, where $r \in_R \mathbb{Z}_q$.

- $\mathcal{O}_{\mathsf{H}_2}$. $\mathcal{B}$ maintains a table $\mathcal{T}_{\mathsf{H}_2} = \{(g^x, \mathsf{H}_2(g^x), r)\}$ to look up the $\mathcal{O}_{\mathsf{H}_2}$-query records. $\mathcal{B}$ takes g^x as input and outputs y if record $(g^x, y, *)$ exists in $\mathcal{T}_{\mathsf{H}_2}$. Otherwise, $\mathcal{B}$ outputs $\mathsf{H}_2(g^x) = g^{\alpha r}$ and inserts $(g^x, g^{\alpha r}, r)$ into $\mathcal{T}_{\mathsf{H}_2}$, where $r \in_R \mathbb{Z}_q$.
- $\mathcal{O}_{\mathsf{Tag}}$. $\mathcal{B}$ maintains a table $\mathcal{T}_{\mathsf{Tag}} = \{(\mathcal{M}, h_{\mathcal{M}}, r, T_{\mathcal{M}}, \Sigma)\}$ to look up the $\mathcal{O}_{\mathsf{Tag}}$-query records. $\mathcal{B}$ takes $\mathcal{M} = (m_1, m_2, \ldots, m_n)$ as input, sets data identifier $h_{\mathcal{M}} = g'^r$, where $r \in_R Z_q$, and chooses tag identifier $T_{\mathcal{M}} \in_R \{0,1\}^\ell$ randomly. For $1 \leq i \leq n$, if $T_{\mathcal{M}}||i$ has been queried to oracle $\mathcal{O}_{\mathsf{H}_1}$, $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ inserts each $(T_{\mathcal{M}}||i, g^{r_i}/h_{\mathcal{M}}^{m_i}, r_i)$ into table $\mathcal{T}_{\mathsf{H}_1}$, where $r_i \in_R \mathbb{Z}_q$, outputs $(h_{\mathcal{M}}, T_{\mathcal{M}}, \Sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n))$, where

  $$\sigma_i = (\mathsf{H}_1(T_{\mathcal{M}}||i)h_{\mathcal{M}}^{m_i})^{sk_{\mathcal{U}}} = ((g^{r_i}/h_{\mathcal{M}}^{m_i})h_{\mathcal{M}}^{m_i})^{\alpha u} = g^{\alpha u r_i} \ ,$$

  and inserts $(\mathcal{M}, h_{\mathcal{M}}, r, T_{\mathcal{M}}, \Sigma)$ into $\mathcal{T}_{\mathsf{Tag}}$.
- $\mathcal{O}_{\mathsf{DK}}$. $\mathcal{B}$ takes user $\mathcal{U}'$'s public key $pk_{\mathcal{U}'} = g^x$ and key token $kt_{\mathcal{U}'}$ as input, looks up whether record $(g^x, *, *)$ exists in table $\mathcal{T}_{\mathsf{H}_2}$, and checks whether $\hat{e}(g, kt_{\mathcal{U}'}) = \hat{e}(g^x, \mathsf{H}_2(g^x))$. If not, $\mathcal{B}$ rejects. Otherwise, $\mathcal{B}$ outputs the delegation key

  $$dk_{\mathcal{U} \to \mathcal{U}'} = \mathsf{H}_2(g^x)^{sk_{\mathcal{U}'}/sk_{\mathcal{U}}} = (g^{\alpha r})^{x/\alpha u} = g^{xr/u} \ .$$

Challenge. After the query phase, $\mathcal{A}$ indicates which $\mathcal{O}_{\mathsf{Tag}}$-query is the target and modifies data to $\mathcal{M}'$. $\mathcal{B}$ looks up the corresponding record in table $\mathcal{T}_{\mathsf{Tag}}$, denoted as ($\mathcal{M}^* = (m_1^*, m_2^*, \ldots, m_n^*)$, $h_{\mathcal{M}^*} = g'^{r^*}$, r^*, $T_{\mathcal{M}^*}$, $\Sigma^* = (\sigma_1^*, \sigma_2^*, \ldots, \sigma_n^*)$), and returns challenge $chal = (C = (c_1, c_2, \ldots, c_n)$, $h_{\mathcal{M}^*}^s$, $H_3(C)^s)$, where $c_i, s \in_R \mathbb{Z}_q$.

Answer. $\mathcal{A}$ returns integrity proof $pf_{chal,\mathcal{V}} = (\rho, V, V', V'', V''')$ using $\mathcal{M}'$. If $\mathcal{M}' \neq \mathcal{M}^*$, we have $V \neq h_{\mathcal{M}^*}^{s\sum_{i=1}^n c_i m_i^*}$ except for a negligible probability. That is, $\Pr[\mathcal{A}$ guesses $\sum_{i=1}^n c_i m_i^* : c_i \in_R \mathbb{Z}_q$ and $\mathcal{M}' \neq \mathcal{M}^*] = \frac{1}{q}$. Otherwise, $\mathcal{A}$ knows the knowledge of $\mathcal{M}^*$[3]. Thus, if $pf_{chal,\mathcal{V}}$ can pass the verification procedure and $\mathcal{M}' \neq \mathcal{M}^*$, we have:

$$\rho^s = \hat{e}(\prod_{i=1}^n \mathsf{H}_1(T_{\mathcal{M}^*}||i)^{sc_i} V, V'')^{sk_{\mathcal{V}}} \tag{6}$$

$$\hat{e}(V, H_3(C)) = \hat{e}(h_{\mathcal{M}^*}, V') \tag{7}$$

$$\hat{e}(V'', g) = \hat{e}(\mathsf{H}_2(pk_{\mathcal{V}}), V''') \tag{8}$$

$$V \neq h_{\mathcal{M}^*}^{s\sum_{i=1}^n c_i m_i^*} \tag{9}$$

[3] $\mathcal{B}$ can extract $\mathcal{M}^*$ by choosing a sequence of linearly independent coefficients adaptively until collecting n valid responses from $\mathcal{A}$. These n linearly independent vectors C_i, $1 \leq i \leq n$, form an $n \times n$ non-singular matrix $[C_1 \ C_2 \ \ldots \ C_n]^{\mathsf{T}} = [c_{i,j}]_{1 \leq i \leq n,\ 1 \leq j \leq n}$. $\mathcal{B}$ uses the $\mathsf{KEA1}$ extractor $\overline{\mathcal{A}}$ to extract the n constant terms $\sum_{j=1}^n c_{i,j} m_j^*$ from V_i and V_i', $1 \leq i \leq n$, and solves the system of linear equations to obtain $\mathcal{M}^*$.

$\mathcal{B}$ can compute $\hat{e}(g, g')^{\alpha^2}$ as follows:

1. Since (7) holds, we have $h_{\mathcal{M}^*}^{\Delta} = H_3(C)$ and $V^{\Delta} = V'$ for some $\Delta \in_R \mathbb{Z}_q$. Thus, $\mathcal{B}$ can use the $\mathsf{KEA1}$ extractor $\overline{\mathcal{A}}$ to extract $m' = \overline{\mathcal{A}}(h_{\mathcal{M}^*}^{s}, H_3(C)^s, V, V')$ such that $V = (h_{\mathcal{M}^*}^{s})^{m'}$. Since (9) holds, we have $m' \neq \sum_{i=1}^{n} c_i m_i^*$.
2. Similarly, since (8) holds, $\mathcal{B}$ can use the $\mathsf{KEA1}$ extractor $\overline{\mathcal{A}}$ to extract $t = \overline{\mathcal{A}}(\mathsf{H}_2(pk_{\mathcal{V}}),\ g,\ V'',\ V''')$ such that $V'' = \mathsf{H}_2(pk_{\mathcal{V}})^t$.
3. After knowing m' and t, since (6) and $m' \neq \sum_{i=1}^{n} c_i m_i^*$ hold, $\mathcal{B}$ can compute $\hat{e}(g, g')^{\alpha^2}$ as follows:

$$
\begin{aligned}
& \rho^s = \hat{e}(\prod_{i=1}^{n} \mathsf{H}_1(T_{\mathcal{M}^*}||i)^{sc_i} V, V'')^{sk_{\mathcal{V}}} \\
\Rightarrow\ & \rho^s = \hat{e}(\prod_{i=1}^{n} (g^{r_i^*}/h_{\mathcal{M}^*}^{m_i^*})^{sc_i} h_{\mathcal{M}^*}^{sm'}, \mathsf{H}_2(pk_{\mathcal{V}})^t)^{\alpha v} \\
\Rightarrow\ & \rho^s = \hat{e}(\prod_{i=1}^{n} (g^{r_i^*}/g'^{r^* m_i^*})^{sc_i} g'^{r^* sm'}, g^{\alpha v' t})^{\alpha v} \\
\Rightarrow\ & \rho = \hat{e}(\prod_{i=1}^{n} (g^{c_i r_i^*}/g'^{r^* c_i m_i^*}) g'^{r^* m'}, g^{\alpha v' t})^{\alpha v} \\
\Rightarrow\ & \rho = \hat{e}(g^{\sum_{i=1}^{n} c_i r_i^*} g'^{r^*(m' - \sum_{i=1}^{n} c_i m_i^*)}, g^{\alpha v' t})^{\alpha v} \\
\Rightarrow\ & \hat{e}(g'^{r^*(m' - \sum_{i=1}^{n} c_i m_i^*)}, g^{\alpha v' t})^{\alpha v} = \frac{\rho}{\hat{e}(g^{\alpha \sum_{i=1}^{n} c_i r_i^*}, g^{\alpha v' t})^v} \\
\Rightarrow\ & \hat{e}(g, g')^{\alpha^2} = (\frac{\rho}{\hat{e}(g^{\alpha \sum_{i=1}^{n} c_i r_i^*}, g^{\alpha v' t})^v})^{1/vv' t r^*(m' - \sum_{i=1}^{n} c_i m_i^*)}
\end{aligned}
$$

$\mathcal{B}$ aborts on handling oracle $\mathcal{O}_{\mathsf{Tag}}$ if tag identifier $T_{\mathcal{M}}$ has been queried to oracle $\mathcal{O}_{\mathsf{H}_1}$. That is, record $(T_{\mathcal{M}}, *, *)$ exists in table $\mathcal{T}_{\mathsf{H}_1}$. For each $\mathcal{O}_{\mathsf{Tag}}$-query, we have $\Pr[(T_{\mathcal{M}}, *, *) \in \mathcal{T}_{\mathsf{H}_1}] = |\mathcal{T}_{\mathsf{H}_1}|/2^{|T_{\mathcal{M}}|} \leq (q_1 + q_T)/2^{\ell}$. Take the union bound on the q_T $\mathcal{O}_{\mathsf{Tag}}$-queries, we have $\Pr[\mathcal{B} \text{ aborts}] \leq (q_1 + q_T)q_T/2^{\ell}$. Moreover, $\mathcal{B}$ loses a negligible portion $\frac{1}{q} = \frac{1}{2^k}$ that $\mathcal{M}' \neq \mathcal{M}^*$ but $V = h_{\mathcal{M}^*}^{s \sum_{i=1}^{n} c_i m_i^*}$. Therefore, the reduced advantage is $\epsilon = (1 - \frac{(q_1+q_T)q_T}{2^{\ell}})(1 - \frac{1}{2^k})\epsilon'$. Besides of handling $(\mathcal{O}_{\mathsf{H}_1}, \mathcal{O}_{\mathsf{H}_2}, \mathcal{O}_{\mathsf{Tag}}, \mathcal{O}_{\mathsf{DK}})$-oracles, $\mathcal{B}$ uses the $\mathsf{KEA1}$ extractor $\overline{\mathcal{A}}$ two times to extract two exponents. Therefore, the reduced time is $t = t' + q_1 t_1 + q_2 t_2 + q_T t_T + q_K t_K + 2t_{\overline{\mathcal{A}}}$. By choosing appropriate $q_1, q_2, q_T, q_K, \ell \in \mathsf{Poly}(k)$, we have $((q_1 + q_T)q_T/2^{\ell}, 1/2^k) \in \mathsf{negl}(k)^2$ and $q_1 t_1 + q_2 t_2 + q_T t_T + q_K t_K \in \mathsf{Poly}(k)$. □

In the proof unforgeability game $\mathsf{Game}^{\mathsf{PF-UF}}$, the challenge *chal* is chosen by the challenger $\mathcal{C}$. $\mathsf{Game}^{\mathsf{PF-UF}}$ can be adapted for existential unforgeability by letting adversary $\mathcal{A}$ choose *chal* by himself. In our security proof, this modification only needs one more execution of the $\mathsf{KEA1}$ extractor to know $\mathcal{A}$'s choice for the randomness s of *chal*.

4.2 Proof Indistinguishability

This game models the notion that a third party without being authorized cannot verify validity of data integrity proofs even if he eavesdrops network communications after the setup phase. In this game, the challenger $\mathcal{C}$ plays the role of the storage server, and the adversary $\mathcal{A}$ plays the role of the third-party user. $\mathcal{A}$ is given access right to oracle $\mathcal{O}_{\mathsf{Proof}}$. $\mathcal{A}$ is trained with valid proofs and tries to verify validity of the target proof. If $\mathcal{A}$ answers validity correctly, he wins this game.

The proof indistinguishability game $\mathsf{Game}^{\mathsf{PF-IND}}$ is as follows:

Setup. $\mathcal{C}$ generates public parameter π, user $\mathcal{U}$'s key tuple $(sk_{\mathcal{U}}, pk_{\mathcal{U}}, kt_{\mathcal{U}})$, delegated verifier $\mathcal{V}$'s key tuple $(sk_{\mathcal{V}}, pk_{\mathcal{V}}, kt_{\mathcal{V}})$, delegation key $dk_{\mathcal{U}\to\mathcal{V}}$, data $\mathcal{M} = (m_1, m_2, \ldots, m_n)$, tags $\Sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$, data identifier $h_{\mathcal{M}}$, and tag identifier seed $T_{\mathcal{M}}$ for $\mathcal{M}$. $\mathcal{C}$ forwards $(\pi,\ pk_{\mathcal{U}},\ pk_{\mathcal{V}},\ kt_{\mathcal{V}},\ dk_{\mathcal{U}\to\mathcal{V}},\ h_{\mathcal{M}},\ T_{\mathcal{M}})$ to $\mathcal{A}$.

Query-1. $\mathcal{A}$ queries oracle $\mathcal{O}_{\mathsf{Proof}}$ to obtain samples of valid proofs.

- $\mathcal{O}_{\mathsf{Proof}}$: $\mathcal{A}$ chooses challenge $chal$ and obtains a valid proof $pf_{chal,\mathcal{V}}$ for $(\mathcal{M}, chal)$.

Challenge. Same as the query-1 phase except that validity of the returned proof $pf^*_{chal,\mathcal{V}}$ depends on an uniform bit b. If $b = 1$, the $\mathcal{C}$ returns a valid proof. Otherwise, $\mathcal{C}$ returns an invalid proof.

Query-2. Same as the query-1 phase.

Answer. $\mathcal{A}$ answers b' for the challenged proof $pf^*_{chal,\mathcal{V}}$. $\mathcal{A}$ wins $\mathsf{Game}^{\mathsf{PF-IND}}$ if $b' = b$. The advantage $Adv_{\mathcal{A}}^{\mathsf{Prf-IND}}$ is defined as $\left|\Pr[\mathcal{A} \text{ wins } \mathsf{Game}^{\mathsf{PF-IND}}] - \frac{1}{2}\right|$.

We show that our scheme is proof indistinguishable under the *truncated decisional* 1-BDHE assumption.

Theorem 2. *If the truncated decisional BDHE problem is $(t, \epsilon, 1)$-secure, the above scheme is $(t - q_1t_1 - q_2t_2 - q_Pt_P, 2\epsilon)$ proof indistinguishable in the random oracle model, where hash functions H_1 and H_2 are modeled as random oracles $\mathcal{O}_{\mathsf{H}_1}$ and $\mathcal{O}_{\mathsf{H}_2}$, (q_1, q_2, q_P) are the numbers of times that an adversary queries $(\mathcal{O}_{\mathsf{H}_1}, \mathcal{O}_{\mathsf{H}_2}, \mathcal{O}_{\mathsf{Proof}})$-oracles, and (t_1, t_2, t_P) are the time used by $(\mathcal{O}_{\mathsf{H}_1}, \mathcal{O}_{\mathsf{H}_2}, \mathcal{O}_{\mathsf{Proof}})$-oracles to respond an oracle query.*

Proof. Let $\mathcal{A}$ be a probabilistic black-box adversary who wins the proof indistinguishability game $\mathsf{Game}^{\mathsf{PF-IND}}$ with advantage ϵ' in time t'. We construct an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to solve the *truncated decisional* 1-BDHE problem as follows:

Setup. Given an instance (g, g^{α}, g', Z) of the *truncated decision* 1-BDHE problem, $\mathcal{B}$ sets the public parameter $\pi = (q, G, g, G_T, g_T, \hat{e}, H_3)$, user $\mathcal{U}$'s key tuple $(sk_{\mathcal{U}},\ pk_{\mathcal{U}},\ kt_{\mathcal{U}}) = (\alpha u,\ g^{\alpha u},\ \mathsf{H}_2(g^{\alpha u})^{\alpha u})$, where $\mathsf{H}_2(g^{\alpha u}) = g^{u'}$ and $u, u' \in_R \mathbb{Z}_q$, delegated verifier $\mathcal{V}$'s key tuple $(sk_{\mathcal{V}},\ pk_{\mathcal{V}},\ kt_{\mathcal{V}}) = (\alpha v, g^{\alpha v}, \mathsf{H}_2(g^{\alpha v})^{\alpha v})$, where $\mathsf{H}_2(g^{\alpha v}) = g^{v'}$ and $v, v' \in_R \mathbb{Z}_q$, and the delegation key $dk_{\mathcal{U}\to\mathcal{V}} = kt_{\mathcal{V}}^{1/sk_{\mathcal{U}}} =$

$g^{vv'/u}$. Then $\mathcal{B}$ chooses data $\mathcal{M} = (m_1, m_2, \ldots, m_n)$, sets data identifier $h_{\mathcal{M}} = g'^{r}$, where $r \in_R \mathbb{Z}_q$, and chooses tag identifier seed $T_{\mathcal{M}}$. For $1 \leq i \leq n$, $\mathcal{B}$ sets $\mathsf{H}_1(T_{\mathcal{M}}||i) = g^{r_i}$, where $r_i \in_R \mathbb{Z}_q$. Then $\mathcal{B}$ invokes $\mathcal{A}$ as a subroutine: $\mathcal{A}^{\mathcal{O}_{\mathsf{H}_1}, \mathcal{O}_{\mathsf{H}_2}, \mathcal{O}_{\mathsf{Proof}}}(\pi,\ pk_{\mathcal{U}},\ pk_{\mathcal{V}},\ kt_{\mathcal{V}},\ dk_{\mathcal{U}\to\mathcal{V}},\ h_{\mathcal{M}},\ T_{\mathcal{M}})$.

Query-1. $\mathcal{A}$ can query oracles $\mathcal{O}_{\mathsf{H}_1}$, $\mathcal{O}_{\mathsf{H}_2}$, and $\mathcal{O}_{\mathsf{Proof}}$ during his execution. $\mathcal{B}$ handles these oracles as follows:

- $\mathcal{O}_{\mathsf{H}_1}$. $\mathcal{B}$ maintains a table $\mathcal{T}_{\mathsf{H}_1} = \{(x, \mathsf{H}_1(x), r)\}$ to look up the $\mathcal{O}_{\mathsf{H}_1}$-query records. $\mathcal{B}$ takes $x \in \{0,1\}^*$ as input and outputs y if record $(x, y, *)$ exists in $\mathcal{T}_{\mathsf{H}_1}$. Otherwise, $\mathcal{B}$ outputs $\mathsf{H}_1(x) = g^r$ and inserts (x, g^r, r) into $\mathcal{T}_{\mathsf{H}_1}$, where $r \in_R \mathbb{Z}_q$.
- $\mathcal{O}_{\mathsf{H}_2}$. $\mathcal{B}$ maintains a table $\mathcal{T}_{\mathsf{H}_2} = \{(g^x, \mathsf{H}_2(g^x), r)\}$ to look up the $\mathcal{O}_{\mathsf{H}_2}$-query records. $\mathcal{B}$ takes g^x as input and outputs y if record $(g^x, y, *)$ exists in $\mathcal{T}_{\mathsf{H}_2}$. Otherwise, $\mathcal{B}$ outputs $\mathsf{H}_2(g^x) = g^r$ and inserts (g^x, g^r, r) into $\mathcal{T}_{\mathsf{H}_2}$, where $r \in_R \mathbb{Z}_q$.
- $\mathcal{O}_{\mathsf{Proof}}$. $\mathcal{B}$ takes challenge $chal = (C = (c_1, c_2, \ldots, c_n), C', C'')$ as input and checks whether $\hat{e}(C', H_3(C)) = \hat{e}(h_{\mathcal{M}}, C'')$. If not, $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ outputs a valid proof $pf_{chal,\mathcal{V}} = (\rho, V, V', V'', V''')$ as below: Let $t \in_R \mathbb{Z}_q$.

$$\begin{aligned}
\rho &= \hat{e}(\prod_{i=1}^{n} \sigma_i^{c_i}, dk_{\mathcal{U}\to\mathcal{V}})^t \\
&= \hat{e}(\prod_{i=1}^{n} (\mathsf{H}_1(T_{\mathcal{M}}||i) h_{\mathcal{M}}^{m_i})^{sk_{\mathcal{U}} c_i}, \mathsf{H}_2(pk_{\mathcal{V}})^{sk_{\mathcal{V}}/sk_{\mathcal{U}}})^t \\
&= \hat{e}(\prod_{i=1}^{n} (g^{r_i} g'^{r m_i})^{c_i}, g^{v'\alpha v})^t \\
&= \hat{e}(g^{\sum_{i=1}^{n} c_i r_i} g'^{r \sum_{i=1}^{n} c_i m_i}, g^{\alpha})^{vv't} \quad ,
\end{aligned}$$

$$V = C'^{\sum_{i=1}^{n} c_i m_i} \ , \ V' = C''^{\sum_{i=1}^{n} c_i m_i} \ , \ V'' = g^{v't} \ , \text{ and } V''' = g^t \ .$$

Challenge. After the query-1 phase, $\mathcal{A}$ chooses challenge $chal = (C = (c_1, c_2, \ldots, c_n),\ C',\ C'')$, and $\mathcal{B}$ checks whether $\hat{e}(C', H_3(C)) = \hat{e}(h_{\mathcal{M}}, C'')$. If not, $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ outputs a proof $pf_{\mathcal{V}}^* = (\rho^*, V, V', V'', V''')$ as follows, where $pf_{\mathcal{V}}^*$ is valid if $Z = \hat{e}(g, g')^{\alpha^2}$. Let $t = \alpha$.

$$\begin{aligned}
\rho^* &= \hat{e}(g^{\sum_{i=1}^{n} c_i r_i} g'^{r \sum_{i=1}^{n} c_i m_i}, g^{\alpha})^{vv't} \\
&= \hat{e}(g^{\sum_{i=1}^{n} c_i r_i}, g^{\alpha})^{vv'\alpha} \times \hat{e}(g'^{r \sum_{i=1}^{n} c_i m_i}, g^{\alpha})^{vv'\alpha} \\
&= \hat{e}(g^{\alpha \sum_{i=1}^{n} c_i r_i}, g^{\alpha})^{vv'} \times \hat{e}(g', g)^{\alpha^2 vv' r \sum_{i=1}^{n} c_i m_i} \\
&= \hat{e}(g^{\alpha \sum_{i=1}^{n} c_i r_i}, g^{\alpha})^{vv'} \times Z^{vv' r \sum_{i=1}^{n} c_i m_i} \quad ,
\end{aligned}$$

$$V = C'^{\sum_{i=1}^{n} c_i m_i} \ , \ V' = C''^{\sum_{i=1}^{n} c_i m_i} \ , \ V'' = g^{\alpha v'} \ , \text{ and } V''' = g^{\alpha} \ .$$

Query-2. Same as the query-1 phase.

Answer. $\mathcal{A}$ answers validity b of $pf^*_{\mathcal{V}}$, and $\mathcal{B}$ uses b to answer the *truncated decisional* 1-BDHE problem directly.

In the above reduction, $\mathcal{B}$ doesn't abort. When $Z = \hat{e}(g, g')^{\alpha^2}$, $\mathcal{A}$ has advantage ϵ' to break proof indistinguishability game. Therefore, the reduced advantage of $\mathcal{B}$ is $\epsilon = \epsilon'/2$ and the reduced time is $t = t' + q_1t_1 + q_2t_2 + q_Pt_P$. By choosing appropriate $(q_1, q_2, q_P) \in \mathsf{Poly}(k)^3$, we have $q_1t_1 + q_2t_2 + q_Pt_P \in \mathsf{Poly}(k)$. □

4.3 Delegation Key Unforgeability

This game models the notion that a third party cannot generate a valid delegation key even if he eavesdrops network communications during the delegation phase and corrupts some delegated verifiers. In this game, the challenger $\mathcal{C}$ provides samples of public keys, key tokens, and delegation keys. The adversary $\mathcal{A}$ corrupts some of the samples to obtain the corresponding private keys and tries to generate a valid delegation key for a user $\mathcal{V}^*$.

The delegation key unforgeability game $\mathsf{Game}^{\mathsf{DK-UF}}$ is as follows:

Setup. $\mathcal{C}$ generates public parameter π and user $\mathcal{U}$'s key tuple $(sk_{\mathcal{U}}, pk_{\mathcal{U}}, kt_{\mathcal{U}})$. $\mathcal{C}$ forwards $(\pi, pk_{\mathcal{U}})$ to $\mathcal{A}$.

Query. $\mathcal{A}$ queries oracle $\mathcal{O}_{\mathsf{Dlg}}$ and oracle $\mathcal{O}_{\mathsf{Cor}}$ to obtain samples of public keys, key tokens, and delegation keys, and the corresponding private keys.

- $\mathcal{O}_{\mathsf{Dlg}}$: It samples a user $\mathcal{V}$ and returns $(pk_{\mathcal{V}}, kt_{\mathcal{V}}, dk_{\mathcal{U}\to\mathcal{V}})$.
- $\mathcal{O}_{\mathsf{Cor}}$: $\mathcal{A}$ chooses $(pk_{\mathcal{V}}, kt_{\mathcal{V}}, dk_{\mathcal{U}\to\mathcal{V}})$ from $\mathcal{O}_{\mathsf{Dlg}}$ and obtains $sk_{\mathcal{V}}$ from $\mathcal{O}_{\mathsf{Cor}}$.

Answer. $\mathcal{A}$ generates a valid delegation key $dk_{\mathcal{U}\to\mathcal{V}^*}$ for a user $\mathcal{V}^*$. $\mathcal{A}$ returns $(sk_{\mathcal{V}^*}, pk_{\mathcal{V}^*}, kt_{\mathcal{V}^*}, dk_{\mathcal{U}\to\mathcal{V}^*})$ to $\mathcal{C}$, where $(sk_{\mathcal{V}^*}, pk_{\mathcal{V}^*}, kt_{\mathcal{V}^*})$ is a valid key tuple for $\mathcal{V}^*$. $\mathcal{A}$ wins $\mathsf{Game}^{\mathsf{DK-UF}}$ if $\mathsf{VrfyDK}(\pi, dk_{\mathcal{U}\to\mathcal{V}^*}, pk_{\mathcal{U}}, pk_{\mathcal{V}^*}) = true$. The advantage $Adv^{\mathsf{DK-UF}}_{\mathcal{A}}$ is defined as $\Pr[\mathcal{A}$ wins $\mathsf{Game}^{\mathsf{DK-UF}}]$.

We show that our scheme is delegation key unforgeable under the InvCDH assumption.

Theorem 3. *If the InvCDH problem is (t, ϵ)-secure, the above scheme is $(t - q_2t_2 - q_Dt_D - q_Ct_C, eq_C\epsilon)$ delegation key unforgeable in the random oracle model, where hash function H_2 is modeled as random oracle $\mathcal{O}_{\mathsf{H}_2}$, e is the Euler's number, (q_2, q_D, q_C) are the numbers of times that an adversary queries $(\mathcal{O}_{\mathsf{H}_2}, \mathcal{O}_{\mathsf{Dlg}}, \mathcal{O}_{\mathsf{Cor}})$-oracles, and (t_2, t_D, t_C) are the time used by $(\mathcal{O}_{\mathsf{H}_2}, \mathcal{O}_{\mathsf{Dlg}}, \mathcal{O}_{\mathsf{Cor}})$-oracles to respond an oracle query.*

Proof. Let $\mathcal{A}$ be a probabilistic black-box adversary who wins the delegation key unforgeability game $\mathsf{Game}^{\mathsf{DK-UF}}$ with advantage ϵ' in time t'. We construct an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to solve the InvCDH problem as follows:

Setup. Given an instance (g, g^{α}) of the InvCDH problem, $\mathcal{B}$ sets the public parameter $\pi = (q, G, g, G_T, g_T, \hat{e}, H_1, H_3)$ and user $\mathcal{U}$'s key tuple $(sk_{\mathcal{U}}, pk_{\mathcal{U}}, kt_{\mathcal{U}}) = (\alpha, g^{\alpha}, \mathsf{H}_2(g^{\alpha})^{\alpha})$, where $\mathsf{H}_2(g^{\alpha}) = g^u$ and $u \in_R \mathbb{Z}_q$. $\mathcal{B}$ invokes $\mathcal{A}$ as a subroutine: $\mathcal{A}^{\mathcal{O}_{\mathsf{H}_2}, \mathcal{O}_{\mathsf{Dlg}}, \mathcal{O}_{\mathsf{Cor}}}(\pi, pk_{\mathcal{U}})$.

Query. $\mathcal{A}$ can query oracles $\mathcal{O}_{\mathsf{H}_2}$, $\mathcal{O}_{\mathsf{Dlg}}$, and $\mathcal{O}_{\mathsf{Cor}}$ during his execution. $\mathcal{B}$ handles these oracles as follows: $\mathcal{B}$ chooses probability $\delta = \frac{q_C}{q_C+1}$.

- $\mathcal{O}_{\mathsf{H}_2}$. $\mathcal{B}$ maintains a table $\mathcal{T}_{\mathsf{H}_2} = \{(g^x, \mathsf{H}_2(g^x), r)\}$ to look up the $\mathcal{O}_{\mathsf{H}_2}$-query records. $\mathcal{B}$ takes g^x as input and outputs y if record $(g^x, y, *)$ exists in $\mathcal{T}_{\mathsf{H}_2}$. Otherwise, $\mathcal{B}$ outputs $\mathsf{H}_2(g^x) = g^{\alpha r}$ with probability δ or outputs $\mathsf{H}_2(g^x) = g^r$ with probability $1-\delta$, and inserts $(g^x, \mathsf{H}_2(g^x), r)$ into $\mathcal{T}_{\mathsf{H}_2}$, where $r \in_R \mathbb{Z}_q$.
- $\mathcal{O}_{\mathsf{Dlg}}$. $\mathcal{B}$ maintains a table $\mathcal{T}_{\mathsf{Dlg}} = \{(v, pk_{\mathcal{V}})\}$ to look up the $\mathcal{O}_{\mathsf{Dlg}}$-query records. $\mathcal{B}$ samples a fresh delegated verifier $\mathcal{V}$, $(pk_{\mathcal{V}}, *, *)$ doesn't exist in table $\mathcal{T}_{\mathsf{H}_2}$, randomly and generates $\mathcal{V}$'s key tuple $(sk_{\mathcal{V}},\ pk_{\mathcal{V}},\ kt_{\mathcal{V}}) = (v,\ g^v,\ \mathsf{H}_2(g^v)^v = (g^{\alpha v'})^v)$ with probability δ or $(sk_{\mathcal{V}},\ pk_{\mathcal{V}},\ kt_{\mathcal{V}}) = (\alpha v,\ g^{\alpha v},\ \mathsf{H}_2(g^{\alpha v})^{\alpha v} = (g^{v'})^{\alpha v})$ with probability $1-\delta$, where $v, v' \in_R \mathbb{Z}_q$. $\mathcal{B}$ inserts $(pk_{\mathcal{V}}, \mathsf{H}_2(pk_{\mathcal{V}}), v')$ into $\mathcal{T}_{\mathsf{H}_2}$ and inserts $(v, pk_{\mathcal{V}})$ into $\mathcal{T}_{\mathsf{Dlg}}$. Then $\mathcal{B}$ outputs $(pk_{\mathcal{V}}, kt_{\mathcal{V}}, dk_{\mathcal{U}\to\mathcal{V}})$, where
$$dk_{\mathcal{U}\to\mathcal{V}} = kt_{\mathcal{V}}^{1/sk_{\mathcal{U}}} = (g^{\alpha v v'})^{1/\alpha} = g^{vv'} \ .$$
- $\mathcal{O}_{\mathsf{Cor}}$. $\mathcal{B}$ takes $(pk_{\mathcal{V}}, kt_{\mathcal{V}}, dk_{\mathcal{U}\to\mathcal{V}})$ as input and rejects if either record $(pk_{\mathcal{V}}, *, *)$ doesn't exist in table $\mathcal{T}_{\mathsf{H}_2}$, record $(*, pk_{\mathcal{V}})$ doesn't exist in table $\mathcal{T}_{\mathsf{Dlg}}$, or $(kt_{\mathcal{V}}, dk_{\mathcal{U}\to\mathcal{V}})$ isn't consistent with $\mathcal{T}_{\mathsf{H}_2}$ and $\mathcal{T}_{\mathsf{Dlg}}$. $\mathcal{B}$ outputs $sk_{\mathcal{V}} = v$ if $pk_{\mathcal{V}} = g^v$. Otherwise, $sk_{\mathcal{V}} = \alpha v$, and $\mathcal{B}$ aborts.

Answer. $\mathcal{A}$ forges a delegation key $dk_{\mathcal{U}\to\mathcal{V}^*}$ for a user $\mathcal{V}^*$ whose key tuple is $(sk_{\mathcal{V}^*},\ pk_{\mathcal{V}^*},\ kt_{\mathcal{V}^*})$, and outputs $(sk_{\mathcal{V}^*}, pk_{\mathcal{V}^*}, kt_{\mathcal{V}^*}, dk_{\mathcal{U}\to\mathcal{V}^*})$. $\mathcal{B}$ rejects if either record $(pk_{\mathcal{V}^*}, *, *)$ doesn't exist in table $\mathcal{T}_{\mathsf{H}_2}$ or $\hat{e}(pk_{\mathcal{U}}, dk_{\mathcal{U}\to\mathcal{V}^*}) \neq \hat{e}(pk_{\mathcal{V}^*}, \mathsf{H}_2(pk_{\mathcal{V}^*}))$. If $\mathsf{H}_2(pk_{\mathcal{V}^*}) = g^{\alpha r^*}$, $\mathcal{B}$ aborts. Otherwise, $\mathsf{H}_2(pk_{\mathcal{V}^*}) = g^{r^*}$, and $\mathcal{B}$ computes $g^{\frac{1}{\alpha}}$ as follows:

$$\begin{aligned}
&\hat{e}(pk_{\mathcal{U}}, dk_{\mathcal{U}\to\mathcal{V}^*}) = \hat{e}(pk_{\mathcal{V}^*}, \mathsf{H}_2(pk_{\mathcal{V}^*})) \\
\Rightarrow\ &\hat{e}(g^{\alpha}, dk_{\mathcal{U}\to\mathcal{V}^*}) = \hat{e}(g^{sk_{\mathcal{V}^*}}, g^{r^*}) \\
\Rightarrow\ &dk_{\mathcal{U}\to\mathcal{V}^*} = g^{sk_{\mathcal{V}^*} r^*/\alpha} \\
\Rightarrow\ &g^{\frac{1}{\alpha}} = (dk_{\mathcal{U}\to\mathcal{V}^*})^{1/sk_{\mathcal{V}^*} r^*}
\end{aligned}$$

In the above reduction, $\mathcal{B}$ doesn't abort with probability $\delta^{q_C}(1-\delta)$. When choosing $\delta = \frac{q_C}{q_C+1}$, we have $\delta^{q_C}(1-\delta) = (1-\frac{1}{q_C+1})^{q_C}\frac{1}{q_C+1} = (1-\frac{1}{q_C+1})^{q_C+1}\frac{1}{q_C} \geq \frac{1}{eq_C}$. Therefore, the reduced advantage is $\epsilon = \frac{\epsilon'}{eq_C}$, and the reduced time is $t = t' + q_2 t_2 + q_D t_D + q_C t_C$. By choosing appropriate $(q_2, q_D, q_C) \in \mathsf{Poly}(k)^3$, we have $(eq_C, q_2 t_2 + q_D t_D + q_C t_C) \in \mathsf{Poly}(k)^2$. □

5 Conclusion

We proposed a delegable provable data possession model that provides delegable (authorized) verification on remote data. Delegable PDP allows a trusted third party to check data integrity under data owner's permission and prevents the trusted third party to re-delegate this verification capability to others. This feature is desired on private data in the public cloud. We provided a construction for the delegable PDP problem and proved its security in the random oracle model.

Due to using pairing operations on blocks directly, each block m_i is limited to k-bit long. We shall develop a new delegation method without using pairing in the future. Dynamic operations, such as insertion, deletion, modification, etc., on stored data is useful. Supporting efficient dynamic operations on stored data is another direction of our future works.

References

1. Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z., Song, D.: Provable data possession at untrusted stores. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, CCS 2007, pp. 598–609 (2007)
2. Ateniese, G., Di Pietro, R., Mancini, L.V., Tsudik, G.: Scalable and efficient provable data possession. In: Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks, SecureComm 2008, pp. 9:1–9:10 (2008)
3. Ateniese, G., Kamara, S., Katz, J.: Proofs of Storage from Homomorphic Identification Protocols. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 319–333. Springer, Heidelberg (2009)
4. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical Identity Based Encryption with Constant Size Ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
5. Boneh, D., Gentry, C., Waters, B.: Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005)
6. Bowers, K.D., Juels, A., Oprea, A.: Hail: a High-availability and Integrity Layer for Cloud Storage. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, CCS 2009, pp. 187–198 (2009)
7. Bowers, K.D., Juels, A., Oprea, A.: Proofs of retrievability: theory and implementation. In: Proceedings of the 2009 ACM Workshop on Cloud Computing Security, CCSW 2009, pp. 43–54 (2009)
8. Chen, B., Curtmola, R., Ateniese, G., Burns, R.: Remote data checking for network coding-based distributed storage systems. In: Proceedings of the 2010 ACM Workshop on Cloud Computing Security Workshop, CCSW 2010, pp. 31–42 (2010)
9. Curtmola, R., Khan, O., Burns, R.: Robust remote data checking. In: Proceedings of the 4th ACM International Workshop on Storage Security and Survivability, StorageSS 2008, pp. 63–68 (2008)
10. Curtmola, R., Khan, O., Burns, R., Ateniese, G.: Mr-pdp: Multiple-replica provable data possession. In: Proceedings of the 2008 the 28th International Conference on Distributed Computing Systems, ICDCS 2008, pp. 411–420 (2008)
11. Damgård, I.B.: Towards Practical Public Key Systems Secure Against Chosen Ciphertext Attacks. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 445–456. Springer, Heidelberg (1992)
12. Erway, C., Küpçü, A., Papamanthou, C., Tamassia, R.: Dynamic provable data possession. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, CCS 2009, pp. 213–222 (2009)

13. Gentry, C.: Practical Identity-based Encryption without Random Oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
14. Juels, A., Kaliski Jr., B.S.: Pors: proofs of retrievability for large files. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, CCS 2007, pp. 584–597 (2007)
15. Shacham, H., Waters, B.: Compact Proofs of Retrievability. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 90–107. Springer, Heidelberg (2008)
16. Wang, C., Wang, Q., Ren, K., Lou, W.: Privacy-preserving public auditing for data storage security in cloud computing. In: Proceedings of the 29th Conference on Information Communications, INFOCOM 2010, pp. 525–533 (2010)
17. Wang, Q., Wang, C., Li, J., Ren, K., Lou, W.: Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing. In: Backes, M., Ning, P. (eds.) ESORICS 2009. LNCS, vol. 5789, pp. 355–370. Springer, Heidelberg (2009)

國科會補助計畫衍生研發成果推廣資料表

日期:2012/05/07

國科會補助計畫	計畫名稱: 異質多網安全檢測平台建置計畫(III)		
	計畫主持人: 謝續平		
	計畫編號: 100-2219-E-009-005- 學門領域: 通訊軟體及平台(網通國家型)		
研發成果名稱	(中文) 分離式的全系統層次模擬器與資訊流動追蹤方法與其應用		
	(英文) Method for Decoupling System-Wide Information Flow Tracking for Malware Analysis and Its Applications		
成果歸屬機構	國立交通大學	發明人 (創作人)	謝續平,王繼偉,劉晏如
技術說明	(中文) 傳統的資安分析與偵測方式中，多偏向以靜態分析的方式觀察這些威脅的外在特徵，如病毒碼或入侵特徵等，但目前進階的攻擊技巧多以執行時期的動態技術，如加殼或變型等，以躲避追蹤。且層出不窮的軟體漏洞，與駭客每天發展出新的攻擊與惡意程式，令資訊安全檢測與鑑識的困難度大大提昇。動態汙染分析為一項極具潛力的技術：透過一軟體模擬的X86系統並實際於其中運行程式的方式，可在執行過程中監控資訊的流動行為。然而動態汙染分析效能與系統模擬彼此交錯執行，將導致記憶體快取命中率過低且令模擬器的指令最佳化失效，因而速度過於緩慢使其可用性降低，且缺乏以整體主機系統作為分析對象的功能，以至於無法分析現今侵入作業系統內部的高階惡意程式。本發明提出一種將系統模擬器與資訊流動追蹤彼此分離的技術，透過此技術令兩個系統能在兩個獨立的CPU平行執行，因此較目前的系統層級的資訊流追蹤系統，執行效率可大幅提高。 備註 ：		
	(英文) System-wide taint analysis is a widely-adopted technique in software testing and malware analysis. To achieve this, a system level emulator equipped with dynamic information flow tracking capability, DIFT, is desirable. However, its effectiveness comes at a price of severe performance degradation due to interleaved system emulation and DIFT analysis. To improve the performance of DIFT, we managed to regain the memory locality and code optimization while executing the interleaved system emulation and taint analysis. In this invention, a new method to decouple DIFT is proposed to parallelize system-wide emulation and taint analysis. The proposed decoupling methods are able to aggressively eliminate dependency between the emulator and the analysis thread. In addition, an encoding method for extracting information flow is also proposed to eliminate the redundant decoding process while preserving byte-granularity accuracy.		
產業別	資訊服務業；研究發展服務業		
技術/產品應用範圍	軟體分析除錯工具, 病毒行為分析		
技術移轉可行性及預期效益	目前提供雲端掃毒服務之公司遍布全球，如知名的防毒軟體公司趨勢科技、卡巴斯基等，皆透過使用者上傳可疑之惡意程式至主機端，進行惡意程式分析後，再提供使用者資訊。本技術所帶來之效能進步，可使DIFT能應用在後端主機進行半即時之掃描服務，因此提供此類服務之公司皆可能應用本發明		

註：本項研發成果若尚未申請專利，請勿揭露可申請專利之主要內容。

100 年度專題研究計畫研究成果彙整表

計畫主持人：謝續平　計畫編號：100-2219-E-009-005-

計畫名稱：異質多網安全檢測平台建置計畫(III)

成果項目			量化			單位	備註（質化說明：如數個計畫共同成果、成果列為該期刊之封面故事...等）
			實際已達成數（被接受或已發表）	預期總達成數(含實際已達成數)	本計畫實際貢獻百分比		
國內	論文著作	期刊論文	0	0	100%	篇	
		研究報告/技術報告	0	0	100%		
		研討會論文	2	0	100%		1.Y.R. Liu, C.W. Wang, J.W. Hsu, S.P. Shieh, ’Extracting Hidden Code from Packed Malware based on Virtual Machine Memory Comparison,’ 21th Cryptology and Information Security Conference (CISC 2011), 2011. 2. C.K. Chen, W.C. Chen, J.W. Hsu, S.P. Shieh, ’Mutant Malware Discovery and Behavior Analysis for Cyber Crime Investigation,’ 22th Cryptology and Information Security Conference (CISC 2012), 2012.
		專書	0	0	100%		
	專利	申請中件數	1	0	100%	件	1.王繼偉，謝續平，劉晏如,’ 分離式的全系統層次模擬器與資訊流動追蹤方法與其應用,’ Taiwan patent pending
		已獲得件數	1	0	100%		S.I. Huang, S.P. Shieh, ’Method and System for Secure

							Data Aggregation in Wireless Sensor Networks, 用於在無線感應器網路中進行安全資料聚合的方法以及系統’ ROC patent no. I350086. 10, 2011.
	技術移轉	件數	1	0	100%	件	建構於行動裝置 ARM CPU 上之污染分析系統，工業技術研究院
		權利金	600	0	100%	千元	
	參與計畫人力（本國籍）	碩士生	19	0	100%	人次	
		博士生	4	0	100%		
		博士後研究員	0	0	100%		
		專任助理	2	0	100%		
國外	論文著作	期刊論文	0	0	0%	篇	
		研究報告/技術報告	0	0	100%		
		研討會論文	0	0	0%		
		專書	0	0	100%	章/本	
	專利	申請中件數	1	0	100%	件	财 CW Wang, SP Shieh, YR Liu, ’Method for Decoupling System-Wide Information Flow Tracking for Malware Analysis and Its Applications,’ US patent pending
		已獲得件數	4	0	100%		财 S.I. Huang, S.P. Shieh, ’Method and System for Secure Data Aggregation in Wireless Sensor Networks 無線傳感器網路中安全數據聚合的方法與系統,’ China patent number ZL200710301500.9, 2011. 财 S.I. Huang, S.P.

							patent no. 8027474, 2011.9.27. 财 Hung-Min Sun, Shih-Pu Hsu, and Chien-Ming Chen, ’ Mobile jamming attack method in wireless sensor network and method defending the same,’ US Patent 7,907,888, March 15, 2011. 财 Hung-Min Sun and Yue-Hsun Lin,’ Pair-wise key pre-distribution method for wireless sensor network,’ US Patent (pending), Application number: 11/599962, Publication number: US 20080044028/A1.
	技術移轉	件數	0	0	100%	件	
		權利金	0	0	100%	千元	
	參與計畫人力（外國籍）	碩士生	0	0	100%	人次	
		博士生	0	0	100%		
		博士後研究員	0	0	100%		
		專任助理	0	0	100%		
其他成果 (無法以量化表達之成果如辦理學術活動、獲得獎項、重要國際合作、研究成果國際影響力及其他協助產業技術發展之具體效益事項等，請以文字敘述填列。)		技術服務【共 1 件】 友訊科技 D-link－委託 Open D-Link Routers Forum 建置、維護與測試技術與諮詢服務 II；無線網路設備開放程式碼網站（社群）建置與安全性分析 產學合作【共 12 件】 調查局－法務部調查局惡意程式自動檢測技術支援系統委託研究採購案 教育部－DNSSEC 推動先期型計畫 中華電信－動態惡意程式行為側錄與汙染分析 中華電信－行動平台資通訊安全問題的研究(二) 中華電信－基於虛擬網路技術適用於異質網路之資源分配最佳化 中華電信－雙階層式全系統汙染鑑識分析 中華電信－行動平台資通訊安全問題的研究(三) 喬鼎科技－前瞻性檔案完整性驗證與可疑嵌入碼檢測平台 工研院－雲端行動的安全及時分析可行性評估先期探討					

	工研院－智慧終端技術研究 工研院－行動終端軟體品質技術研究 宏達電子(HTC)－雲端惡意程式鑑識與行動平台安全（含先期技轉） 教育部－DNSSEC 網域名稱安全架構建置與推廣計畫 趨勢科技--Technology Transfer on Network Threat Detection using Security Log Correlation

🟥	成果項目	量化	名稱或內容性質簡述
科教處計畫加填項目	測驗工具(含質性與量性)	0	
	課程/模組	0	
	電腦及網路系統或工具	0	
	教材	0	
	舉辦之活動/競賽	0	
	研討會/工作坊	0	
	電子報、網站	0	
	計畫成果推廣之參與（閱聽）人數	0	

國科會補助專題研究計畫成果報告自評表

請就研究內容與原計畫相符程度、達成預期目標情況、研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性)、是否適合在學術期刊發表或申請專利、主要發現或其他有關價值等，作一綜合評估。

1. 請就研究內容與原計畫相符程度、達成預期目標情況作一綜合評估

■達成目標

□未達成目標（請說明，以 100 字為限）

□實驗失敗

□因故實驗中斷

□其他原因

說明：

2. 研究成果在學術期刊發表或申請專利等情形：

論文：□已發表 ■未發表之文稿 □撰寫中 □無

專利：□已獲得 ■申請中 □無

技轉：■已技轉 □洽談中 □無

其他：(以 100 字為限)

3. 請依學術成就、技術創新、社會影響等方面，評估研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性)（以 500 字為限）

在 2011 年，本計畫開發了 7 個全新的安全檢測防護工具，且繼續客制化與維護已開發完成的 15 個檢測工具。同時，我們也持續把適切的檢測工具轉成線上服務，讓更多人可因此受惠。藉由此平台的建置與檢測工具的開發，我們希望提供政府機關、財團法人及高科技廠商網路安全檢測的服務，並且技轉所開發的檢測工具，以幫助上述單位發現漏洞及弱點。如此一來將可提高產業的經濟效益、提升無線產品附加價值、節省因網路攻擊或系統弱點所消耗的產值、節省專業檢測人力並且有效減少各種有線、無線網路環境的攻擊。