Information Security Final Project

A Formal Analysis of IEEE 802.11’s WPA2: Countering the Kracks Caused by Cracking the Counters Cas Cremers, Benjamin Kiesl, and Niklas Medinger, CISPA Helmholtz Center for Information Security

<https://www.usenix.org/conference/usenixsecurity20/presentation/cremers>

古佳儫 60947005S

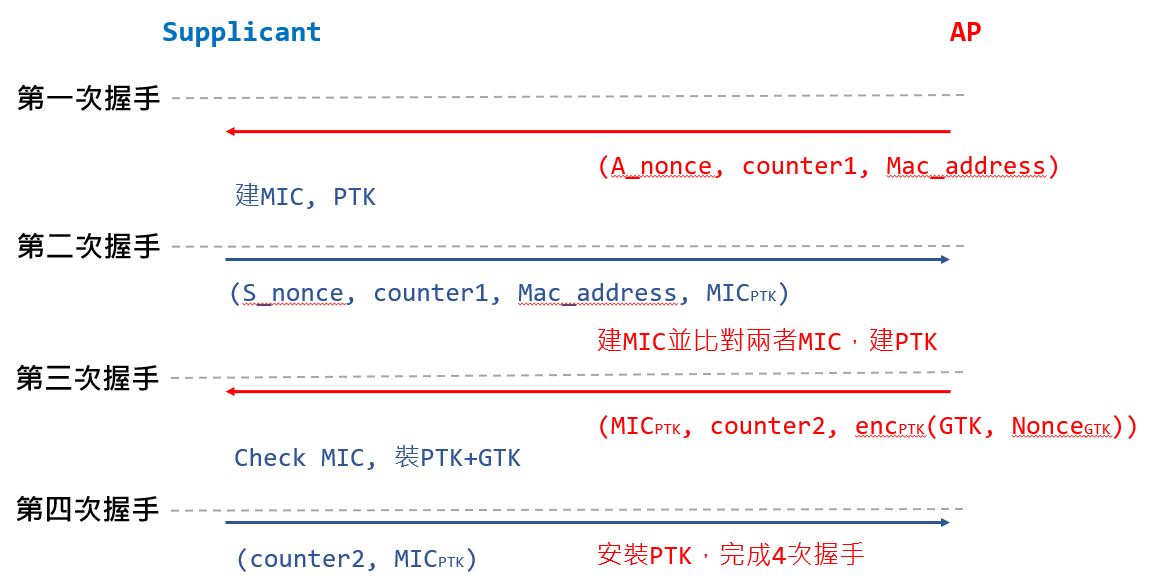
鄭博升 60947038S

**研究動機：**

這次期末報告，起初想要找一篇可以實作的paper來實際玩玩看，在看到這篇paper的標題是對wpa2的krack攻擊分析時，基於實作過中間人攻擊的作業後對此類型的題材非常有興趣，於是毫不猶豫的決定可以來研究看看，由於我們在作業中嘗試進行中間人攻擊時的先備知識的不足，很多東西都是從頭開始查到尾，然而老師上課也有介紹到wpa2以及krack攻擊，讓我們對於wpa2稍微有了一些認識。但在讀完後才知道這篇paper其實是針對目前的wpa2協定進行詳細分析、證明並且設計一個安全實用的模型，其中驗證與證明的部分是使用第一作者所設計的Tamarin Prover來進行，實際下載作者的模型，並將其導入Tamarin後發先其內容驚為天人的複雜，但也充滿了規則，每個元件都在重重lemma下一步一步進行自動驗證。在經過這篇paper探討後，我們也藉此更了解目前市面上我們使用的wpa2是如何運作的，並且存在著哪些風險需要注意，最後雖然沒有完整的實作可以讓我們嘗試，但我們透過作者設計的模型在Tamarin上的驗證中學到很多新知。

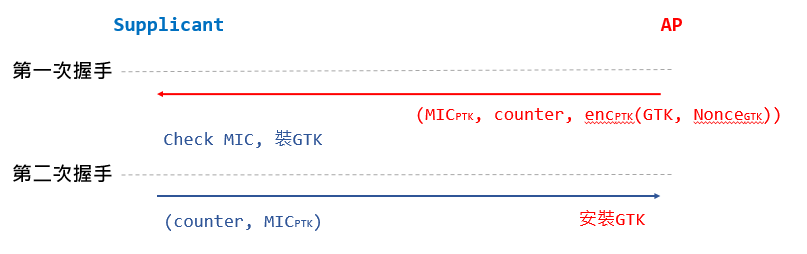
**背景介紹：**

Four-way Handshake



當supplicant想要與AP端建立連線，會先輸入wifi密碼，此時會產生PMK(由WIFI密碼、SSID經過hash function所構成)，接著AP會對supplicant發送A\_nonce、counter1、Mac\_address，此為第一次的握手，而counter是作者設計的Replay Counter利用receiver必須回覆和sender訊息中所挾帶的counter數值相同的數值來避免replay attack，當supplicant收到訊息後，即可透過雙方的Mac\_address建立MIC，並透過雙方的nonce與PMK建立PTK，而supplicant會將S\_nonce、counter1、Mac\_address、MIC回傳給AP端，完成第二次握手，在AP收到後，也會用雙方的MAC\_address建立MIC，並比對傳來的MIC是否一致，來確保封包的完整性，接著也會建立PTK，並傳送MIC、counter2以及被PTK加密的GTK，做為第三次握手，supplicant則會在確認MIC正確後，安裝PTK與GTK，並回傳counter2、MIC，告知AP安裝完成後，此時Four-way handshake就完成了。

Group-key Handshake



GTK是一個群組暫時金鑰，主要用於區網內的廣播訊息加密，因此在一段時間或是有supplicant請求更換GTK時，AP端會透過Group-key Handshake的方式來發送新的GTK給所有連線中的supplicant，運作模式類似於four-way handshake中的第三與第四次握手。

**實驗模型：**

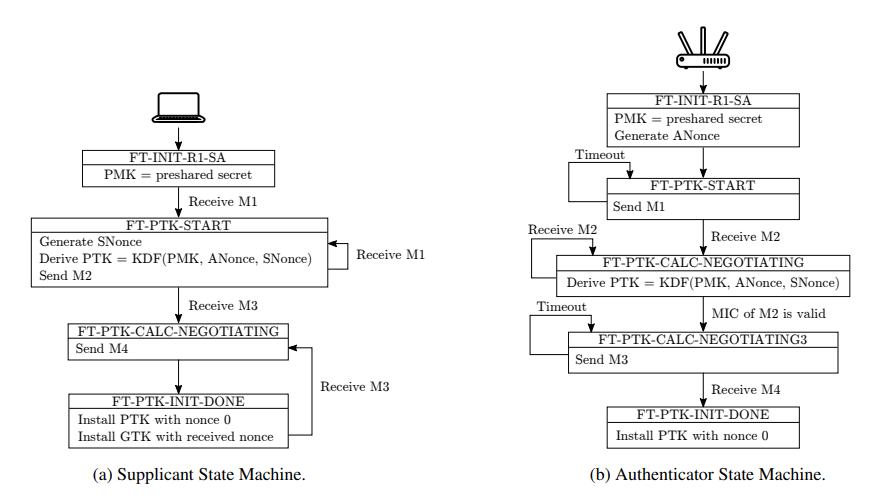
作者透過Tamarin Prover建立four-way handshake與group key handshake等模型，並透過此軟體進行自動驗證。

在wpa2中，若要實作four-way handshake與group-key handshake須滿足以下規範

* 實作four-way handshake需使用至少兩台的狀態機，一台為supplicant，另一台為AP
* 實作group-key handshake需使用至少兩台狀態機，一台為supplicant，另一台為AP
* 一台狀態機專門用於產生新的group key

Four-way handshake

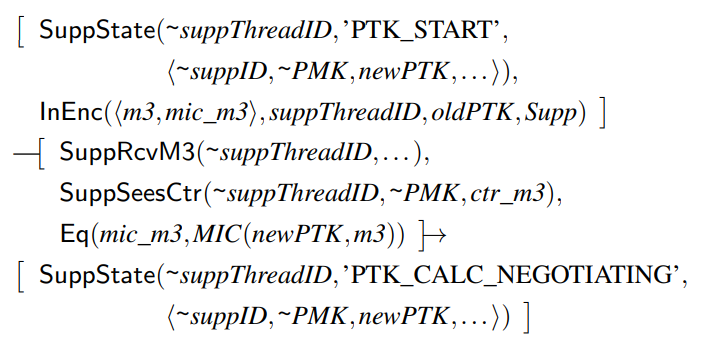
Four-way handshake 狀態轉換圖



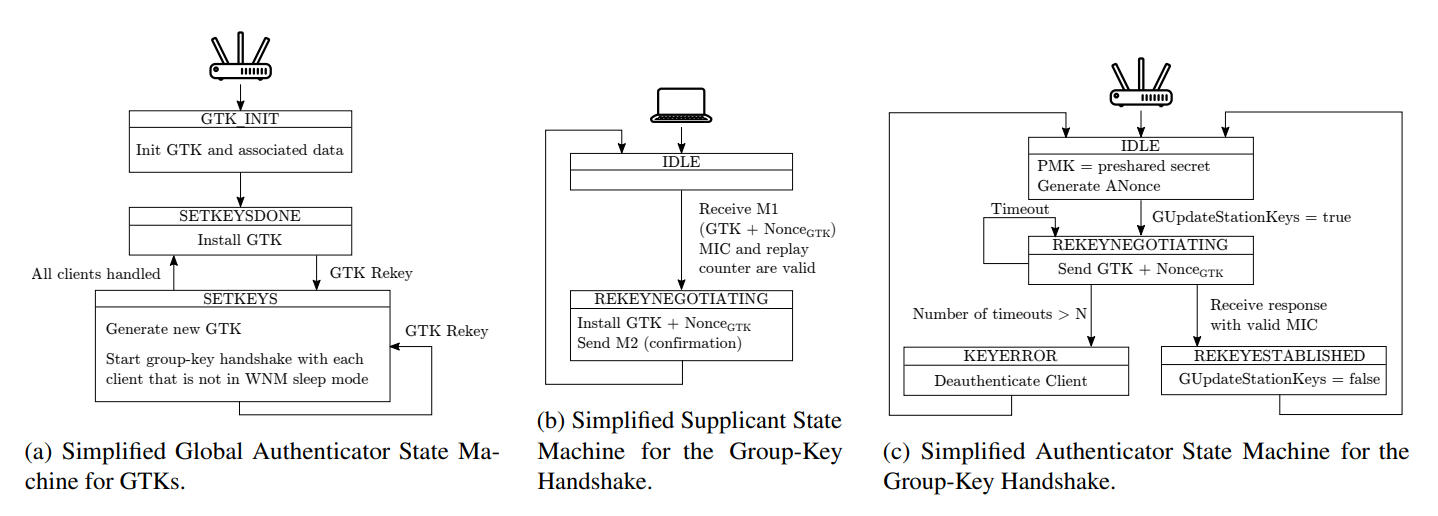
在four-way handshake中，當傳送完M1以及M3後，遲遲沒有收到supplicant的訊息，則會造成timeout，導致AP端重新傳送一次原訊息，這也造成了之後的krack問題。

[腳本1](#腳本1)作者展示了supplicant端，從FT-PTK-START狀態轉換到FT-PTK-CALC-NEGOTIATING狀態的Tamarin程式碼。SuppState代表現在的狀態在PTK\_START，並且每個STA都有一個獨一無二的suppThreadID，"~"代表之前生成的東西，InEnc代表從網路中接收到的資訊，SuppRcvM3在此model中用於驗證lemmas，SuppSeesCtr 用於通過限制對replay counter機制進行建模，Eq 用於確保此message的MIC是valid

腳本 1: FT-PTK-START狀態轉換到FT-PTK-CALC-NEGOTIATING狀態



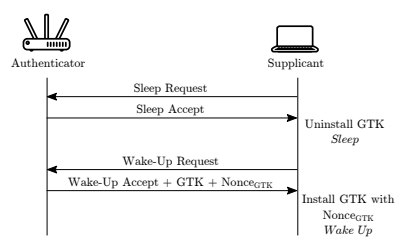
Group-key handshake

Group-key handshake狀態轉換圖

在group-key handshake中，(a)用於生成新的GTK，(b)與(c)則用於模擬group-key handshake，每個supplicant可透過發送型態為group bit的Request bit設為1的EAPOL-key frame 觸發group key handshake，AP 也可在有人斷線或認證失敗的時候發起group-key handshake，以及從 sleep mode 醒來的supplicant也需要進行group-key handshake來拿到當前使用的GTK。

在此model中，four-way handshake與group-key handshake無法同時發生，若同時都需要進行，則以four-way handshake優先，並且在handshake過程中的replay counter是相對第一個訊息中各自的handshake replay counter，因此這兩種handshake也不該同時進行。

WNM Sleep Mode

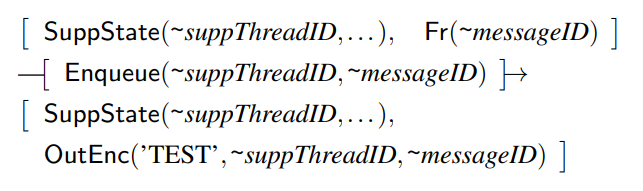


此模式允許supplicant進入睡眠模式來節省能源，此時會將GTK解除安裝，並不參與group-key handshake，此處作者為了使模型更加具有一般性，讓WNM相關的通訊模式只要在建立好PTK後，可以和其他handshake同時發生。另外提到再進入睡眠模式後必需將原先安裝的GTK刪除，才不會有後續Message Queue的問題產生。

Message Queue

在真實情況中，很有可能發生訊息已送出，但卻還在Message queue排隊的現象，所以假定在一種情景是supplicant申請新的GTK但有為時間的關係訊息還在Queue中排隊，但在尚未申請完成前，如果supplicant使用GTK加密時就可能造成使用舊的GTK的情況發生，因此作者設計一個[腳本2](#腳本2)來實作Message Queue，並透過Tamarin Prover來驗證這類的漏洞不會在他的系統發生。在此模型中，每個supplicant都有獨立的一個message queue，和自己的threadID，圖中的OutEnc代表將訊息內容放入message queue中。

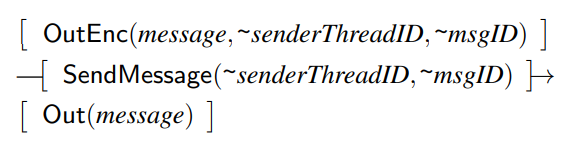
腳本 2：Message Queue



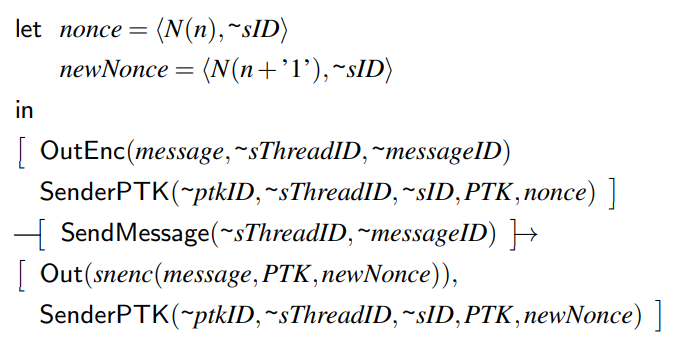
WPA2加密與傳送

WPA2的訊息加密採用AESCCMP來加密，[腳本3](#腳本3)在尚未建立PTK，訊息將直接以明文方式傳送，[腳本4](#腳本4)在建立PTK後，則是會有一個初始化的nonce搭配AES加密產生key stream，再與明文做XOR，並且每次加密完後nonce的值會一直往上加，以避免nonce reuse而產生Krack的漏洞。

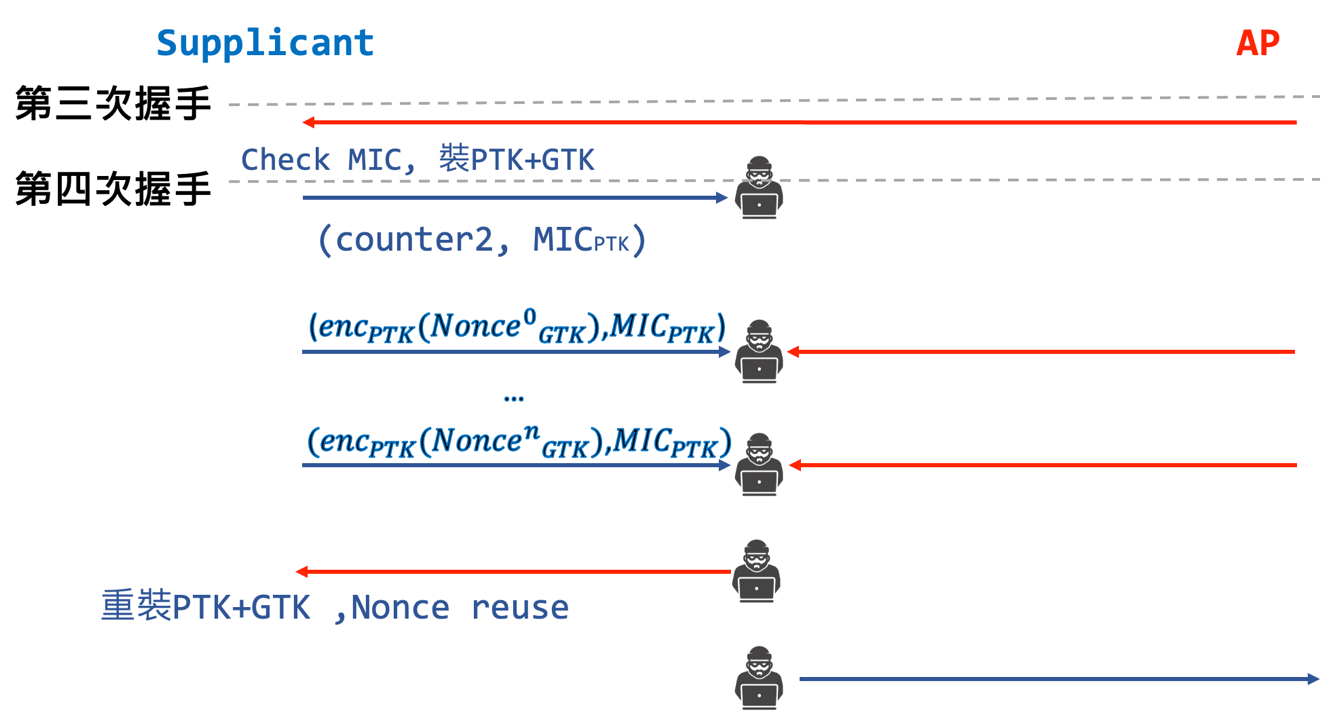
腳本 3：PTK尚未安裝



腳本 4：PTK安裝後

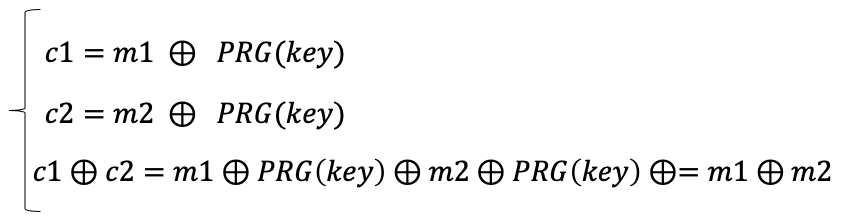


**Krack:**

****

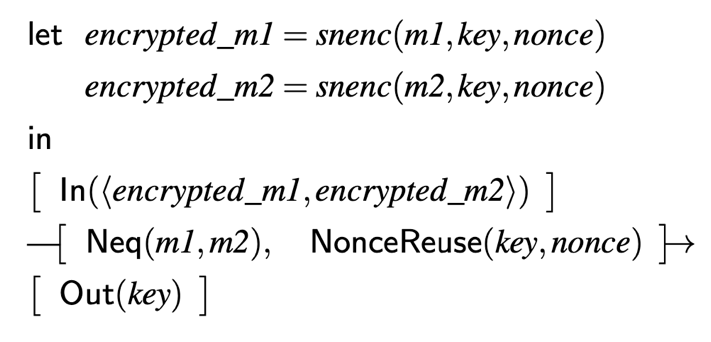
1. 一般情況下的Krack發生在第四次握手時，supplicant在傳送ACK給AP時，如果中間攻擊者劫持著這段訊息，則AP在未收到supplicant的ACK時會覺得supplicant還未正確安裝PTK+GTK，於是將會再等待某一個時間timeout之後，重新發送第三次握手的訊息給supplicant，然而supplicant在收到重複的第三次握手的訊息時，就算已經安裝好PTK+GTK了，依然會因為是既有的設計而重複安裝PTK+GTK的動作，經過反覆操作，造就了中間攻擊者可以在中間進行控制，因為當supplicant在安裝完PTK+GTK後，會根據CCMP加密流程中的取用Nonce來和TK進行AES加密的這個動作，在未修補的情況下Nonce都是經由初始值零至某個上限值，而且TK都是同一個的情況下，如果中間攻擊者可以在經過n次後取得重複的Nonce，那就代表nonce reuse的情況發生，這時候就會非常像Broken one-time-pad的情況，同一個key stream重複使用造成中間攻擊者可以利用[公式1](#公式1)的作法，即使不需要知道完整的key stream也可以暴力破解出敏感資訊。

公式 1: Broken one-time-pad

****

1. 在發生最壞的情況下，如果發生Krack且中間攻擊者可以透過nonce reuse的計算找出完整的Key Stream那不就代表作者可以再往後supplicant所傳輸的訊息直接被洩露，如圖表1所示。作者針對[worst case design](#腳本5)的情況在Tamarin上並透過nonce reuse 不會發生的lemma，來確保這個最壞的情況不會發生在作者所設計的模型中。

腳本5：worst case design

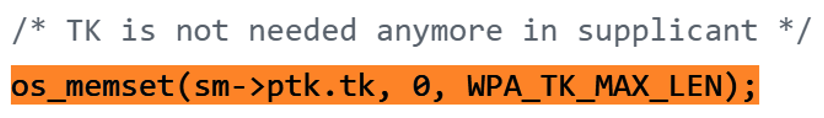


圖表1：worst case的假設情況，Key Stream被完整洩漏

|  |  |  |
| --- | --- | --- |
| **worst case** | | |
| Nonce | Key | Key Stream |
| 0000 | f4b7 |  |
| 0001 | f4b7 |  |
| 0002 | f4b7 |  |
| .. | f4b7 | .. |
| ffff | f4b7 |  |
| 0000 | f4b7 |  |

1. Krack如果發生在Linux & Android 2.4/2.5 version system時會發生一個特殊的情況，工程師在設計時因為沒想到會發生Krack，於是在supplicant安裝完PTK+GTK後立刻就把TK歸零，從wpa未修正前的開源[程式碼](#程式碼1)找到確實有一段是當TK不再需要時就將其歸零的設計，這個動作導致後續如果發生重裝PTK+GTK這件事情的時候，supplicant會使用全為零的TK來生成Key Stream，於是往後supplicant在通訊時都會以這個不安全的Key Stream，所以通訊一直都處在極大的風險之中，但這個問題其實並沒有被作者提及，而且問題也在軟體更新後解決。

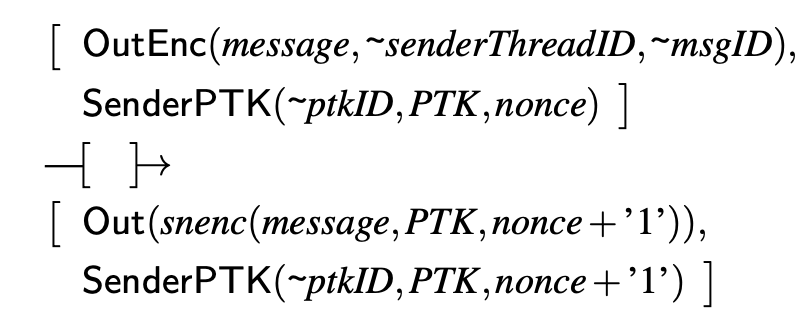
程式碼 1



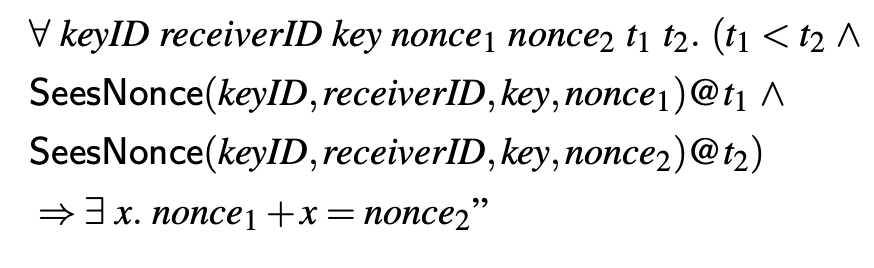
**Prevent Krack：**

[Replay counter rule](#腳本6)：針對CCMP加密時的nonce做一個間單的設計，讓nonce reuse這件事情不會發生。作者採用起始值是一個非零的任意數值以nonce稱之，並確保每次在進行加密的過程中所使用的nonce都比前一個值還要大至少一個單位的規則來設計，並且引用Mathy Vanhoef and Frank Piessens. Key reinstallation attacks: Forcing nonce reuse in WPA這篇paper的策略: supplicant在安裝完PTK+GTK後不可擅自更改或重置CCMP加密所使用的nonces，換句話說只有你在重新進行four-way handshake時才會將nonce重置成某個任意值。再透過設計在Tamarin的[檢查腳本](#腳本7)，每次都對於下一個加密所使用的nonce檢查必定是在大於某個數值x的情況下，才能確保nonce reuse的情況不存在。

腳本6：Replay counter rule

****

腳本7:nonce reuse check rule

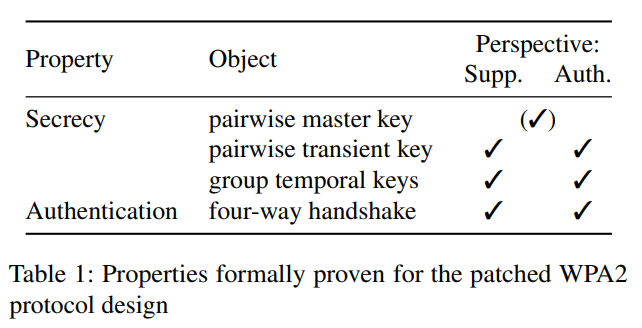
****

**Kr00k Vulnerability：**

ESET 2020年在資安大會上發佈的關於博通和聯發科製作wifi晶片上的漏洞，其漏洞在於針對wifi解離時緩衝區中的封包因為時間的關係還未被傳送完全，但此時卻因為晶片設計會將暫時金鑰設成零來對那些封包進行加密傳輸，所以中間攻擊者如果可以控制supplicant和AP之間解離時，中間攻擊者就可以在敏感的時間取得敏感的資訊，然而這個漏洞在發布後幾個月博通及聯發科都分別發佈wifi驅動程式來修補這個漏洞，其中修補的方式並未在網路上公佈，這邊推測可能是當supplicant和AP發生wifi解離的狀況時，針對那些還在緩衝區的封包就不再進行任何加密傳輸，而是直接丟棄，就可以解決這個問題。

**Conclusion:**

作者針對模型的安全性以用四大項目分別來證明其是否安全，其中PTK、GTK、Four-way handshake更是以supplicant和AP端兩個不同的角度來進行分析：

****

**PMK:**

PMK由passphrase、SSID經過hash function的結果，由於PMK並未透過網路直接傳輸，而且在加密上野並非直接使用PMK來做加密，所以對於PMK的安全只要wifi密碼未被洩露，那PMK的安全就無須考慮。

**PTK:**

考慮到PTK是由PMK、Anonce、Snonce經過一系列運算生成的，所以在只要在PMK不被洩漏的情況下那PTK也應該就是安全，並且作者已經利用reply counter證實他的模型不會有nonce reuse問題，所以PTK是安全的。

**GTK:**

因為GTK是由PTK加密而來，所以如果GTK是不安全，那就表示PTK已經被洩漏，又因為PTK是由PMK、Anonce、Snonce經過一系列運算生成的，因此若PTK被洩漏那表示PMK也被洩漏，但不失一般性假設PMK不會被洩漏，而且model證明nonce reuse不可能會發生所以PTK也是安全的且GTK也是安全。

**Four-way handshake:**

由於Four-way handshake只會執行一次，且Anonce與Snonce每次生成都不一樣且傳輸過程中都只有一個，基於這樣的前提下作者在Tamarin中所設計的一系列lemma，來證實Four-way handshake有滿足injective agreement。

總結來說作者針對WPA2在基於Krack的攻擊為其研究動機提出了一個能抵擋各種可能出現攻擊的模型架構，並透過工具Tamarin來進行理論和數學上的驗證，也確實驗證抵擋包括Krack各種假設攻擊。