訊息追追追　真假照妖鏡：哼哈HASH&MAC

◆ 社團法人台灣 E 化資安分析管理協會理事長、中央警察大學資訊密碼暨建構實驗室（ICCL）— 王旭正教授

我們在 Security 的概觀中曾提過，鑑定（Authentication）、鑑識（Forensics）、密碼元件等元素，在 Security 的資安領域中一定要知道一些密碼的概念，Security 才能做得好。在現在的網路時代裡，真的假的容易混淆，分不太清楚，這使得 Security 的「鑑定」，或者「鑑識」，就變得格外的重要！資訊領域裡，我們經常看到這個字眼「Cyber」，Cyber 翻譯成中文的意思還不錯，一般通稱為「網際空間」。數學裡的空間可以是二維、三維、多維空間；那在網路裡就是全球性的概念了。若想將科技層次拉高，可以在科技用詞前面加個「Cyber」這個詞，例如「Cyber Security」，「Cyber Forensics」。現在許多科技與資安的主題也會加個「Cyber」，似乎就水漲船高，成了全球性無所不包的資訊科技議題了。

鑑識，我們說過就是找出蛛絲馬跡。在現代科技網路發達的時代，訊息互相流通是如此迅速，電腦、手機「一陽指」操作，按下傳送即會不經意地傳送到任何地方，速度之快令人咋舌，真假之間許多事都被假戲真作了！我們在上一期中，提到發布一個消息之後，在公開金鑰系統下可

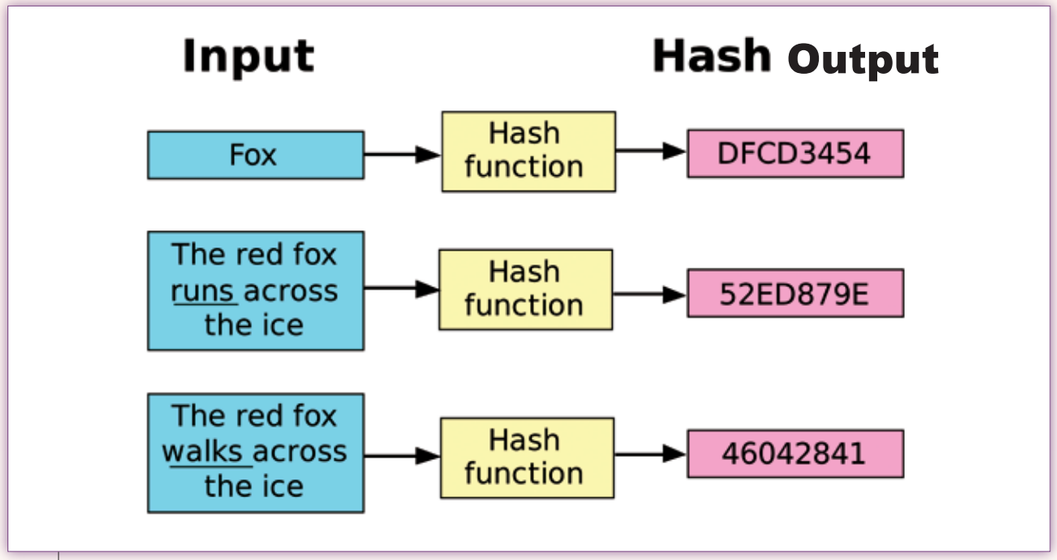
網際空間是全球性的概念，現代科技發達，訊息互通迅速，電腦、手機「一陽指」操作按下傳送，即會不經意地將訊息傳送到任何地方。

用 HASH 將訊息做處理，接著搭配發送者的祕密金鑰產生驗證碼，與訊息一起在網路傳送。收到訊息的人若想知道真假，可使用發送者的公開金鑰做運算並做驗證碼的比對，就可以判斷真假訊息了！

是的，假訊息的判定在資安科技裡可以運用密碼學的概念做處理，得能還原真相，不需流於「口水戰」。實質上的技術層面有幾個可以處理的方法。其中一個是以「公開金鑰」系統的概念去處理，即用密碼學裡公開金鑰系統的「數位簽章（Digital Signature）」技術進行真假訊息

假訊息可透過密碼學裡公開金鑰系統的「數位簽章（Digital Signature）」技術來判定。

判讀（也就是我們前期文章提到的方式）；另一個是可用「HASH」的技巧。若因為訊息長度很長，可以用 HASH 轉換成比較短的長度，HASH 甚至可以保證：若這個訊息遭到更改，在 HASH 的運作裡可看的清清楚楚。



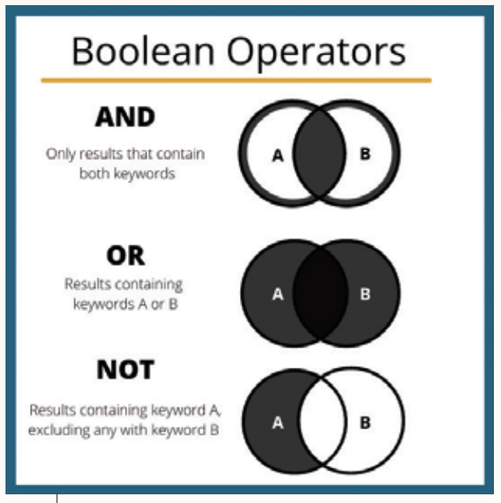
HASH 可把大量的資料變小後再依後續資安的需求做處理。

HASH

這個 HASH 雖然神奇，卻也平易近人、出身平凡呢！在我們談資訊安全的過程當中，依需求有時候要將訊息做加密的「保護」或簽章的「鑑定」處理。但如果是一本書我們要做加密／簽章的處理，因為每個字都很重要，所以不能只處理書裡面一部分的內容，因為你認為不重要的也許他人覺得很重要！所以書裡面的內容全部都要做處理、保護，這時候就需要全部做 HASH，不能厚此薄彼。這就是 HASH 存在的價值，不論資料量多大，都可以透過 HASH 把資料量變小然後再依後續資安的需求做處理，讓運作非常有效率。

為什麼 HASH 這麼神奇，可以把大量的資料變得那麼短小呢？讓我們想像一下，若一個訊息有 1,000 公尺這麼長，要將它變得很短，例如變成 100 公尺，那是不是可以把每 100 公尺剪成一段一段，剪成 10 段，將這些都疊起來，那原 1,000 公尺的訊息不就可以縮成只剩 100 公尺的長度！那有沒有什麼運算的方法，可以讓他們疊起來還是 100 公尺呢？而且要精準，不能差一絲一毫。就像搬家的時候，冰箱比門大一點點，就是沒有辦法搬進去，所以要懂得變通。變通之一，把門的螺絲卸掉，拆掉門，然後把冰箱搬進去，再把門組裝回來。

那麼在資訊科技裡，想想什麼運算可以這樣呢？回到剛才所談的 HASH，要將 10 段 100 公尺的訊息疊起來，還是 100 公尺的長度，用加法可以嗎？加法不行；乘法可以嗎？乘法更誇張，長度會變得更大。這是很有趣的狀態。另類思維裡，我們來思考一下，訊息拆成一段一段，加的不行，乘的不行，有一個運算叫「OR」，也有「AND」，還有一個叫「NOT」，這 OR、AND、NOT 是布林（Boolean）運算裡的邏輯運算基本「三兄弟」。這三兄弟的邏輯運算子還可以變出另外兩種：「XOR」與「XNOR」，得以加速電腦的運算速度。



布林運算裡的基本「三兄弟」：OR、AND、NOT。（Photo Credit: Cecelia Vetter, https://commons.wikimedia.org/wiki/File:Diagram_Explaining_Boolean_Operators.png）

這裡我們看到雖然加減乘除在我們生活中很有用，但是在做 HASH 時卻派不上用場。因為在電腦中是數位型態存在，所以各訊息的加與乘運算會增加訊息的長度，行不通的。然邏輯運算的 OR、AND、NOT，以及它們的變化 XOR 與 XNOR，卻反而發揮最大的效果。也就是說，當將長度相同的訊息做邏輯運算時，並不會增加原訊息的長度，這項特質造就了 HASH 的「神奇」。回顧剛剛說到的「一個訊息有 1,000 公尺長」，若目標為縮短為「1 公尺」，那麼就將每 1 公尺剪成一段一段，即能裁剪成 1,000 段，將這些都疊起來，相疊裡的運算都採用邏輯運算，那麼原 1,000 公尺的訊息不就可以準確地濃縮成所設定目標的「1 公尺」長度，變得更短了。至此，是否覺得 HASH 雖是神奇但觀念簡單而平易近人呢！

HASH 的原理這麼簡單，那 HASH 的種類有那些呢？ HASH 並不是只有一種，就像這世界上的汽車難道只有「TOYOTA（豐田）」這種品牌的汽車嗎？當然還有「FERRARI（法拉利）」品牌的汽車（跑車）。在 HASH 的模式與基本原理下，當然可有許多製作的方式／品牌，而 HASH 的製作演算法就有「MD5」還有「SHA」。

演算法名稱	輸出大小 (bits)	內部大小	區塊大小	長度大小	字元尺寸	碰撞情形
HAVAL	256/224/192/160/128	256	1024	64	32	是
MD2	128	384	128	No	8	大多數
MD4	128	128	512	64	32	是
MD5	128	128	512	64	32	是
PANAMA	256	8736	256	否	32	是
RadioGatún	任意長度	58字	3字	否	1-64	否
RIPEMD	128	128	512	64	32	是
RIPEMD-128/256	128/256	128/256	512	64	32	否
RIPEMD-160/320	160/320	160/320	512	64	32	否
SHA-0	160	160	512	64	32	是
SHA-1	160	160	512	64	32	有缺陷
SHA-256/224	256/224	256	512	64	32	否
SHA-512/384	512/384	512	1024	128	64	否
Tiger (2) -192/160/128	192/160/128	192	512	64	64	否
WHIRLPOOL	512	512	512	256	8	否

HASH 有多種製作演算法，其中，MD5 在 2004 年被分析出資安破解疑慮後失去優勢。（Photo Credit: WIKI, https://zh.wikipedia.org/wiki/%E6%95%A3%E5%88%97%E5%87%BD%E6%95%B8）

例如，若我們希望 HASH 最後輸出長度是「128」，基本概念下可想像將原輸入訊息每 128 的長度切一段，然後依照我們所說明的方式通通疊起來做邏輯運算，一旦原訊息總長度 1,000，不是 128 的倍數，最後那段不足 128 的部分將會技巧性做填補到 128 的長度而得以一起做堆疊式的邏輯運算，當然無庸置疑也造就 HASH 最後長度是 128，訊息瞬間變短了。

此外，HASH 還具有單向函數的特質，也就是說，輸出的短訊息無法逆推回原來所輸入的較長訊息。如同一塊玻璃碎了沒有辦法再全部修補回去，回不去了。HASH 的處理過程透過邏輯運算，由長變短，最後的輸出結果，專業術語即為「DIGEST」。先前我們提到 HASH 的製作演算法有多種，例如 MD5、SHA 還有 TIGER。其中較通用的是 MD 系列的 MD5，而 MD5，在 2004 年被分析出資安破解疑慮，雖有做些強化，但也似乎失去優勢了，藉此開始了 SHA 的舞臺。MD5 有資安疑慮後，SHA 系列即強化設計機制而有更新的演算法。MD5 與 SHA 的基本比較上，MD5 的輸出，DIGEST 長度是 128 位元，SHA 的輸出 DIGEST 長度規格有 160 的基本款，也有擴充版能使得 DIGEST 的長度到達 256、384、512 等位元。

HASH 函數的功能與相關性質，整理如圖 1 所示。圖 1 中將不同類型的訊息，經由 HASH 函數的運算之後，可以得到一組固定長度的短訊息，「DIGEST」。HASH 函數的運算具有的三種特殊性質，分別是「單向性」、「抗碰撞」與「擴張性」。其中「單向性」指的是只能得到右邊的輸出結果但是無法反推回去，如同汽機車單行道一樣，所以叫單向；「抗碰撞」指的是不同的字有不同的對應輸出結果，不會出現不同的文字卻有相同對應輸出的情形；「擴張性」指的是即使只是一些微小的文字變化，而會得到大為不同的輸出結果。由圖 1 我們可以發現，儘管輸入的內容僅僅為「空 /null」、「1」及「2」等訊息上的差異，但經由 HASH 函數所產生的 DIGEST 可以很明顯的看出所輸出的結果有相當大的差異。

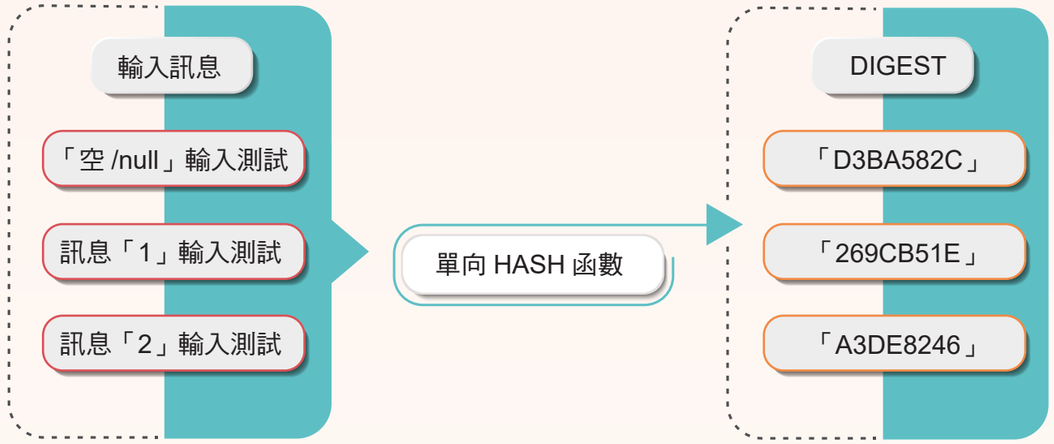


圖 1　HASH 函數的 DIGEST 輸出

判斷消息的真假，可以透過公開金鑰系統，也可以透過以上提到的 HASH。但是操作公開金鑰系統的代價是每個人彼此都要有公開金鑰的事先處理設定，如果沒有，就無法處理假訊息。HASH 當然也是處理假訊息的利器，也因 HASH 的特質是可公開取得，所有欲判斷訊息的人皆可直接使用 HASH 做比對來得知訊息的真假。此外，在同一個工作、生活圈，甚而軍事、特殊用途時，能否運用公開金鑰系統與 HASH 判斷假訊息的優勢，又不需要做公開金鑰的操作設定，即可輕鬆地完成假訊息的判讀呢？有的，資安密碼的「MAC（Message Authentication Code）」呼之欲出得以勝任此一需求與趨勢。

MAC

MAC 不但可以處理假訊息，也不需要每個人都先設定公開金鑰系統。由於在生活、工作共同活動的群體環境中，得相互擁有一個共同的 Key 是正常的理念，如同在一個辦公室的工作環境進出同一個門，同辦公室人員能有共同的 Key 得以開鎖進入辦公室。MAC 對於同一群體，諸如同事、同袍、好朋友之間，可以用很輕鬆的方式來判讀訊息，防止假訊息的散播。

MAC 的運作與 HASH 相當類似，能將很長的訊息轉化成很短的資料長度，並搭配驗證碼的比對得以判斷真假訊息，兩者最大的不同是在過程中，MAC 會在訊

息裡多加發送訊息者的 Key。若以 HASH 為例說明 MAC 的設計，MAC 就是在處理 HASH 的過程裡，在訊息中（前面、中間、後面都可以）放入了 Key。依我們提到 HASH 的特質，加入 Key 的訊息所產生的新輸出將會明顯地不同於未有 Key 的原 DIGEST 輸出。

那為什麼 MAC 會跟訊息的鑑定／鑑識有關？試想若將訊息加上共同群體的 Key，作 HASH 運算後所得到的 DIGEST 為「驗證碼」，然後將此「驗證碼」放在訊息的最後面，當作是驗證碼，隨著訊息一起傳送。當群體內的人員收到訊息後以驗證碼再去做比對，就立即能判讀訊息的真假。事實上，MAC 這個機制在軍事、醫療情資等特殊用途上是有效率、好用且非常重要的運作機制。例如，在戰事的群體通訊中，同陣營的兩方通訊傳遞，倘使中間過程敵方陣營製造假訊息，由於同陣營裡成員間有了共同的 Key，就可以精確地判斷出真假訊息。

有了「鑑識」的概念，「真」的假不了，「假」的在「火眼金睛」裡立即鑑識出真偽。我們以圖 2 裡「孫悟空（老孫）」、「牛魔王（老牛）」、「芭蕉公主（小芭）」與「白骨精（小白）」為例，將 HASH 與 MAC 的搭配做說明。同一通訊群組裡的「孫悟空」、「牛魔王」與「芭蕉公主」具有共同的 Key。一旦妖精，例如小白欲傳「假」訊息給老孫，由於小白非通訊群組裡的人員，故沒有共同的 Key，當小白欲以老牛或小芭的名義傳送消息給老孫，老孫看到訊息，欲知「真」或「假」，將先產生驗證碼。在 HASH 運作裡，由於小白是自編的 Key，而老孫使用通訊群組共同的 Key 產生新驗證碼，在驗證碼的交叉比對下，藉由比對結果的成立與否即能迅速判斷出小白所傳送是「假」訊息。

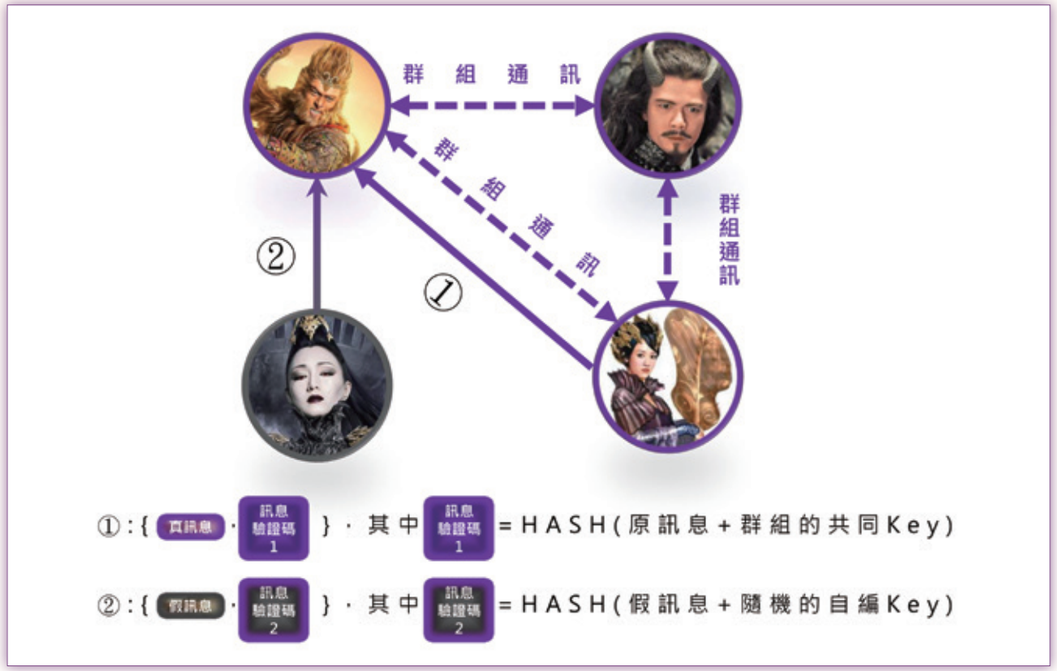


圖 2　HASH 與 MAC 驗證碼的真假訊息判斷

在「鑑識」的世界裡，「HASH」與「MAC」這兩位左右護法、哼哈二將的搭配，讓「鑑識」有如神助，輕易地追出訊息的真假。面對資訊時代，網路裡穿梭往返的各式訊息，「鑑識」的意識培養、「資安」與「密碼」的環環相扣，無疑地是抵禦、判斷真假訊息的資訊科技時代最重要推手，也才得讓資安生活，心（內心的思維）與形（訊息的形式）合而為一，得能掌握資料、資訊、知識的正確判讀、汲取與傳播，而享受科技、相信科技。

社團法人台灣 E 化資安分析管理協會（ESAM）

中央警察大學資訊密碼暨建構實驗室（ICCL）