可容錯之鍊結獨立混合型多媒體認證法

林芬蘭[1]
靜宜大學資訊工程學系
e-mail:lan@pu.edu.tw

林士正[2]
靜宜大學資訊管理學系
e-mail:f9121212@hotmail.com

摘要

此論文提出一個可容錯之鍊結獨立混合性多媒體認證法，並與相關鍊結性多媒體認證法比較。本方法在每個封包內放置二個認證碼，一個認證封包本身，另一個則以鍊結性方式認證其前一封包。當鍊結性認證失敗時，即進行封包獨立認證來判定錯誤的封包，以進行重送或略過。本方法僅在每個封包多出一個認證碼空間與湊雜運算之時間以及些許的封包編號密碼製作時間的付出下，具有一般簡易鍊結性認證法快速認證的之優點，並解決其無容錯力之缺失；同時也因以封包編號為獨立驗證碼之關鍵值而具有影音格式之相容性。

關鍵字:BAFV、DFA、HCPI

1. 介紹

在多媒體內容的認證上，傳統文獻上已有很多的方法，例如可以依每個封包進行簽章，然後將每個封包的認證碼以一個 table 記錄，再將 table 傳遞給接收端，而接收端就以此 table 中之認證碼進行所接收多媒體內容的認證，此方法的缺點是 table 所佔的空間過大，所需要的前置時間亦過長。為改善所需的空間與時間，有學者提出了鍊結性認證的方法-Backward Authentication and Forward Verification (BAFV)[8]。BAFV 之簽章方法是將每一個封包透過如 MD5、SHA-1[11][12]等雜湊運算出的驗證值放入前一個封包的保留空間中;其認證是比對每一個封包的雜湊值與存放於前一個封包中的驗證值，因此在時間與空間的需求都很經濟;同時當整個封包被換掉或遺失時，會因封包資訊鍊結性的破壞而察覺；然而，也因此導致後續所有封包無法繼續驗證。

事實上，多媒體內容傳輸，常會有封包的錯誤或遺失，因此能即時發現錯誤或遺失的封包以便重送或略過，而繼續認證是很重要。因此有學者提出了 Double Forward Authentication (DFA)之驗證法[4]。DFA 方法為 BAFV 的變化，主要是放置雜湊運算的動作進行二次，如將封包 1 的認證碼放入封包 2、3 中，若封包 2 的認證碼與封包 1 之雜湊值不相符，可以立即與封包 3 進行比對。DFV 雖改善 BAFV 在任一封包驗證失敗即無法繼續驗證之缺點，但當連續二個封包錯誤時，仍無法繼續認證。Augmented chain (AG)[7]亦是將認證碼放置二次的認證方法，AG 將整個多媒體串流分成由數個相同簽章順序的小單元組成，每個小單元中的每個封包之雜湊值存放於二處，一處於相鄰之封包，一處於相隔數個封包之後，以降低數個封包連續遺失或錯誤而導致無法繼續認證之風險。Butterfly authentication scheme[9]是另一個將認證碼放置二次的認證方法，將封包編列成多層次等封包數的架構，簽章時，第一個雜湊值放置於每個層次中相同位置的前後層次鍊結上，第二個雜湊值放

置於前後層次中 2 的冪次方之不同位置上。此方法可以依簽章圖簡易進行規則性的簽章，而每個層次中封包的雜湊值放置不同層次的不同位置上，可以降低鄰近封包同時錯誤無法繼續認證之風險。除了多放置一個雜湊值來增加容錯性外，亦有學者提出隨機放置雜湊值於數個不同的封包中以增強封包容錯率，如方法 [1]。

在多媒體內容的認證上亦有學者提出適用於一般影音編碼方式的認證方法。2-D BAFV[5][6]是先將多媒體封包依時間切分成基礎層與數個增強層。其簽章由最末時間之串流開始自最高增強層逐一簽章至基礎層之封包。認證則由接收之基礎層逐一往上驗證，此法本質上與 BAFV 相同因此在容錯性上並無增強。Content-aware 認證法[10] 是配適 JPEG2000 編碼方式的認證法，此法可依既定的影像品質決定認證碼的最佳數量。另外亦有學者提出格式相容的認證法[2][3]，把內文的認證碼放置於 engine 上，在進行認證時，將 engine 上之認證碼取下進行比對。

以雜湊值進行單一之封包獨立認證法無法察覺整個封包被換掉的狀況；而現有之鍊結性認證法則會有因其中某一或數個封包的遺失或錯誤而無法繼續進行認證之容錯力不足的缺點。因此，本論文提出一個植基於湊雜鍊結與封包 ID 獨立認證之容錯認證法(Hash Chain and Packet ID Authentication (HCPI))，在每個封包內放置二個認證碼，一個認證封包本身，另一個則以 BAFV 方式認證其前一封包。當鍊結性認證失敗時，則進行封包獨立認證來確定錯誤的封包，以進行重送或略過錯誤的封包並繼續認證。因此 HCPI 僅在每個封包多出一個認證碼空間與湊雜運算之時間以及些許的封包編號密碼製作時間的付出下，具有 BAFV 快速認證的之優點，並解決其無容錯力之缺失。相較於其他部份容錯驗證法 HCPI 具有 100%之封包認證能力。

本論文的結構如下，第二節介紹相關之認證方法，第三節詳細介紹植基於封包 ID 與湊雜鍊結之容錯認證法 HCPI，接著在第四節是與相關方法的比較，最後進行討論及結論。

2. 相關方法

- BAFV(Backward Authentication and Forward Verification) [8]

BAFV 為鍊結性認證的典型方法，假設一個多媒體串流被封裝成 T 個 packet 其簽章法是由時間 t=T 開始，將每個 packet 之 hash 值存放於其前一個 packet 內，直至 t=1 止而認證則由時間 t=1 開始，依次以存放於下一個 packet 內之認證碼與以接收的 packet 所計算出之 hash 值比對，圖 1 為此方法之示意圖。

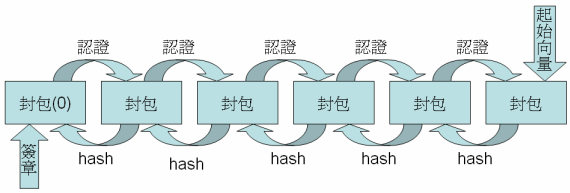

圖 1. BAFV 示意圖

BAFV 演算法:

簽章:

1. 由 t=T 開始，let packet'$_T$ =<packet $_T$,v0> 其中 v0 為常數
2. 選定一湊雜函數 H 並計算 h(T)=H(packet'$_T$)
3. For t= T-1 to 1
   packet'$_t$ =< packet'$_t$, h(t+1)> ，其中 h(t)=H(packet'$_t$)
4. h0=<h(1),T,m,m_0> ，其中 m 是 packet

的長度，m_0 是 H(・)的長度

5. 產生一個額外的 packet'$_0$=<h0, sign(h_0,K)>，sign 是以金鑰 K 的一個簽章函式
6. 最後形成可驗證的新串流 packet'$_0$ packet'$_1$,…, packet'$_T$

驗證:

1. 由接收之 packet$^*_0$，取出 h^*_0 以計算 sign(h^*_0,K) 將之與 h^*_0 驗比較 若相等則繼續
2. t=1 開始至 t=T
   由接收之 packet$^*_{t-1}$ 取得 packet$^*_t$ 之正確 hash 值 $h^*(t)$
   接收 packet$^*_t$ 並計算其 hash 值 H(packet$^*_t$)
   若 H(packet$^*_t$)= $h^*(t)$，繼續下一個

- Double Forward Authentication (DFA)[4]

DFA 方法為 BAFV 的變化，主要是將 packet 之雜湊值存放於緊鄰的前/後兩個 packet 中，如將封包 1 的認證碼放入封包 2、3 中，此方法，若封包區塊 2 出現錯誤，可以立即往下一個進行比對，當連二個封包區塊錯誤則無法繼續向下認證，因此 DFA 方法的容錯率不高。

- Augmented chain [7]

將認證碼放置於二區形成兩個鍊結，一為 local links，作為相鄰封包認證碼放置；另一為 global links，作為跨越數個封包之認證碼放置。整個多媒體內容由數個小單元所組成，而每個小單元則由認證一個封包所需要的封包所組成。假設一小單元有 3 個封包 1、2、3，其中封包 2 的認證碼放置於封包 1、3 中，意即所有的封包認證碼放置於前後封包內，此即完成 local link；接著要將此小單元加入 global links，此時將此小單元前一及後一單元內的相鄰封包認證碼放置於此單元內的封包 1、3 中，而此單元的封包 1、3 認證碼則放置於前後單元的相鄰封包中，即完成小單元加入多媒體序列中。

- Butterfly authentication scheme [9]

將所有的封包進行如圖 2 的排序，假設封包數為 n，N 表每個 stage 的封包數，算出 n=N($\log_2^{N+1}$)，其中 $\log_2^{N+1}$ 之值表 stage 數，接著進行認證碼放置，其規則為將所有前後相鄰 stage 內相同的位置封包放置一次認證碼，如 stage3 第一個位置封包之認證碼放置 stage2 第一個位置封包內，接著進行前後相鄰 stage 內不同位置封包的認證碼放置，且每個 stage 使用跨越封包數需為 2 的冪次方，並依 stage 遞增，如 stage3 第一個位置封包認證碼放置於 stage2 第二個位置封包內，而 stage2 第二個位置封包認證碼放置於 stage1 第四個位置封包內，並於最後 stage0 內所有的封包進行 sign。

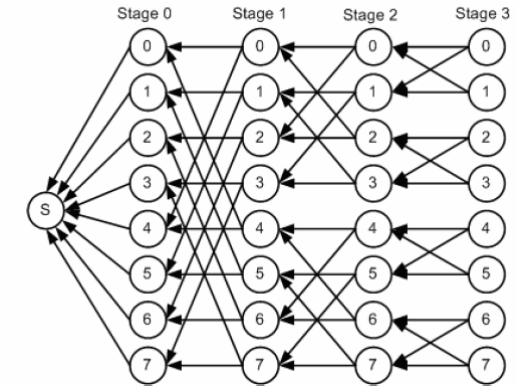

圖 2. Butterfly authentication graph 一例

- JPEG2000 簽章方式

一些學者進一步將鍊結性認證方法進行變化，透過認證的順序改變達到不同於傳統鍊結性認證的簽章方式，如將需要認證的位元流透過封包區塊的切割，使得每個封包區塊中所含認證碼數量將不一定相

同，亦即含多個認證碼的封包區塊為需要且必要取得之封包區塊，不含或只含一個認證碼之封包區塊為非絕對必要取得之封包區塊，意指含認證碼封包內會有多個認證碼(雜湊值)，分別認證不同封包區塊，而其認證碼封包亦具有本身之認證碼可能放置於下一個含認證碼封包區塊中，其示意圖 3 解釋如下:

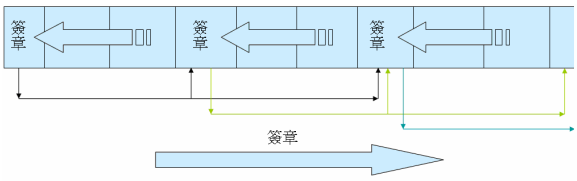

圖 3. 示意圖

現有 10 個封包要進行鍊結性認證，若以 BAFV 方式進行認證，其每個封包區塊將含有上一個封包區塊的認證碼，相反之，並非都將所有封包進行同一方向的鍊結性認證，如上圖，封包區塊 2, 3 及封包區塊 5, 6 和 8, 9 都先進行一次雜湊運算，並將其值放入封包區塊 1, 4, 7 中，接著才將封包區塊 1, 4, 7 進行鍊結性認證，於此情況下，當接收端進行驗證時，封包區塊 2, 3, 5, 6, 8, 9，若有錯誤則不影響整個鍊結性認證的過程。

然而配適於 JPEG2000 為基礎之認證方式，其示意圖 4 如下:

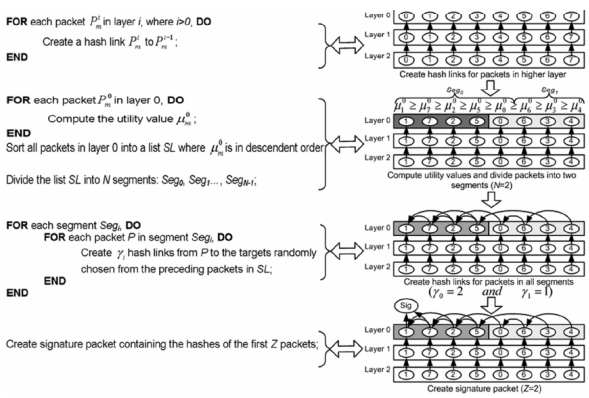

圖 4. JPEG2000 簽章示意圖

漸進式的影像編碼方式，由基層(base)開始，一層一層的增加其影像品質，如上圖中，將其分為 3 層(Layer0~2)，首先將每一影像區塊對應的高層至低層進行 hash links，接著將封包分為邏輯區塊，並且使其邏輯區塊對邏輯區塊都有 hash links，最後進行起始簽章，於此認證方式上，Layer0 的封包都將需要優先取得，接著進行 Layer0 的邏輯區塊認證，若認證無誤，則依序從 Layer1~2 的封包認證，Layer1~2 的封包區塊因接收端的不同需求而有不同的影像封包需求，意即非必要取得，而 Layer0 是基層為必須取得，在整個鍊結性過程中，所有接收端都將進行 Layer0 的鍊結性認證，接著，再因需要進行 Layer0 以上之高層對低層的鍊結性認證方式，透過不同的鍊結性認證順序，而使其每個封包區塊將含不同的認證碼數量，含多個認證碼數量者為重要且必要取得封包區塊，含少數或不含認證碼封包區塊者，為非必要取得且可以容錯的封包區塊。

3. 可容錯之鍊結獨立混合性多媒體認證法

以雜湊值進行單一封包獨立認證會衍生整個封包被換掉而無法察覺的問題，而鍊結性認證雖可解決單一封包獨立認證的缺失，卻會因為其中某一或數個封包的遺失或錯誤而無法繼續認證，譬如說，假若封包 n 錯誤，則可能是封包 n 本文錯誤，或是其內含之鍊結封包之認證碼錯誤。若是認證碼錯誤 則可以確認封包 0 至封包 n-2 為正確，而封包 n-1 之正確與否則無法判定；又錯誤若非認證碼而是本文，則可以確定封包 n-1 亦正確。但由於封包 n 錯誤，所以在比對封包 n+1 內之認證碼時會產生錯誤，故無法繼續進行鍊結性認證。

可容錯之鍊結獨立混合性多媒體認證法，簡稱 HCPI，是結合 BAFV 鍊結性認證與以封包編號為關鍵值之封包獨立認證的認證法。簽章時在每個封包內放置二個認證碼，一個認證封包本身，另一個則以 BAFV 方式認證前一個封包。認證時，首先進行鍊結性認證，當鍊結性認證失敗時，則進行封包獨立認證來確定錯誤的封包。

當資料經由 packet switching network 傳送至接受端前，都會先進進行實體層的封包封裝，將資料區分成許多的封包，並給予每個封包唯一的編號。HCPI 之封包獨立認證法即是以這唯一的編號作為封包獨立認證的基礎。HCPI 認證法詳細說明如下

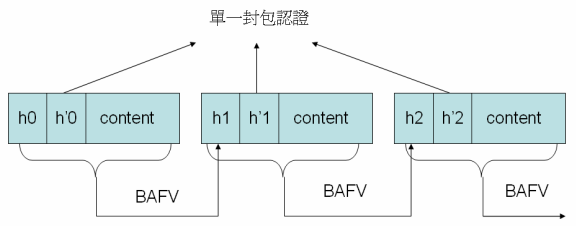

圖 5. HCPI 認證法示意圖

如圖 5 中，每個封包 p_i 存放二個認證碼 h_i 與 h_i'，其中 h_i'用以認證封包本身，而 h_i 則用以認證前一封包，前面述及封包獨立認證是以具唯一性之封包編號為基礎，然而封包編號通常是連續的，缺乏隱密性，因此封包編號必須先經過秘密的轉換後始得以被採用來計算封包獨立認證碼以下先敘述 HCPI 之封包編號秘密轉換法-PKRND(Packet-number Randomization)以及樹狀結構- tree(b, t)

PKRND: 簡單來說，是將封包編號先轉換成樹狀編號，再由樹狀編號對應成密碼簿編號

樹狀結構 tree(b, t)：依封包總總數 n 來決定，其中 b 為分支數、t 為階數(不含 root) $\log_b n >= t$，圖 6 為將 9 個封包分置成 2 階(不含 root)，3 分支之樹狀結構。

PKRND 演算法:

Input: 封包總數 n、封包編號、密碼集、樹狀結構 tree(b, t)

Output: 封包編號密碼

1. 以 PKID 演算法將封包編號轉換成樹狀編號，如圖 6 所示，編號 0 封包轉換成樹狀編號 00，編號 1 封包轉換成樹狀編號 01，其餘封包之樹狀編號類推。
2. 將密碼集分為 b 個子密碼集，並給予每個子密碼集編號 0 至 b-1
3. 依封包樹狀結構編號所對應之子密碼集中分別取出密碼依序串聯後設定為該封包之編號密碼，如編號 1 封包之樹狀編號為 01 則其編號密碼為編號 0 子密碼與編號 1 子密碼之串聯。

封包樹狀編碼演算法 PKID(b, t, n):

Input: n: packet number, b: number of tree branches, t: height of tree (不包含 root)。

Output: packet number in tree format r_t, $r_{t-1},\ldots,r_1$

Let i=1, x = n-1

while i<= t

do

x div b =(q_i, r_i)

x= q_i

i=i+1

End while

Return (r_t, $r_{t-1},\ldots,r_1$) ;

HCPI 驗證法可由發送端與接收端分別來說明，其演算法如下:

發送端

前置作業

1. 發送端隨機產生一本密碼集，並利用公開金鑰加密方式傳遞這本密碼集。
2. 建立數狀結構 tree(b, t)
3. 將密碼集分為 b 個子密碼集，並給予每個子密碼集編號 0 至 b-1

對每個封包簽章:

4. 以 PKRND 演算法將封包編號轉換成樹狀編號密碼。
5. 將每個封包的編號密碼進行雜湊運算，將算出之雜湊值放入自己封包中的 h'欄位。
6. 將每個封包(h=0‖ h'‖content)進行雜湊運算，運算出的認證碼再一次覆蓋於 h' 欄位。(h 原是存放前一封包之驗證碼，因本法是認證單一封包本身,故令 h=0 來排除其它封包之影響)

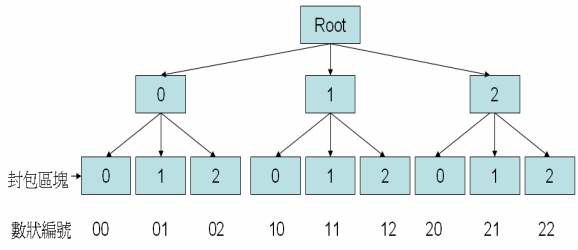

圖 6. 封包編號轉換成樹狀編號示意圖

接收端

前置作業

1. 發送端隨機產生一本密碼集，並利用公開金鑰加密方式傳遞這本密碼集。
2. 建立數狀結構 tree(b, t)
3. 將密碼集分為 b 個子密碼集，並給予每個子密碼集編號 0 至 b-1

封包認證:

For（封包 i=1;i<n+1;i++）

1. 進行 BAFV 鍊結性認證

a. 若 h_i= H(0…0‖h_{i-1}'‖$content_{i-1}$)則繼續進行下一封包之鍊結性認證

b. 若 h_i≠H(0…0‖h_{i-1}'‖$content_{i-1}$)則表示封包 i 或封包 i-1 有誤

2. 進行封包 i 之獨立認證

a. 以 PKRND 演算法將封包編號轉換成樹狀編號密碼。

b. 自封包中取出其獨立認證碼 h_i'* 並令 temp= h_i'*

c. 重新算出編號密碼雜湊值 h_i''。

d. 重新計算 h_i'=H(0…0‖ h_i''‖ $Content_i$'*)。

e. 比對 h_i'與 temp 值，若相同則表示此封包正確 否則即可判定此封包有誤。

3. 若確定封包 i 無誤，則表示封包 i-1 之 content 有誤，此時可要求自封包 i-1 重送或放棄封包 i-1，繼續進行下一封包之鍊結性認證

4. 若確定封包 i 有誤 則可要求自封包 i 重送或放棄封包 i 繼續進行下一封包之鍊結性認證

4. 系統分析與討論

4.1 各認證方法之比較

本節將我們所提出的 HCPI 認證法與前述相關方法依 Zhang 等在[12]之比較項目進行比較，比較項目如下:

- Communication overhead:指多媒體內容認證時，除本文內容外，需要額外的空間供認證碼或簽章值之放置。
- Verification/容錯 probability: 認證過程中，在封包遺失或錯誤發生時，仍可以繼續完成鍊結性認證的可能性。
- Computation overhead:指發送端傳送多媒體內容至接收端之前，發送端需要進行簽證及雜湊運算的時間。
- Sender delay:在第一個多媒體內容封

表一　多媒體鍊結性認證方法比較

	BAFV	DFA	augmented chain	butterfly	HCPI
Communication overhead	s +nh	s +2nh	s +2nh	s +2nh	2nh
Verification probability	no	var	var	var	100%
Computation overhead	ℓ+nt	ℓ+nt	ℓ+nt	ℓ+nt	2nt
Sender delay	n	n	p	n	n
Receiver delay	1	1	n	1	1

包可以傳送之前，在發送端的封包延遲。

- Receiver delay:在第一個多媒體封包可以被認證及解碼之時，在接收端的封包延遲。

說明:

表一為我們的方法 HCPI 與相關鍊結性認證法針對上述項目之比較結果，其中 n 表封包總數、h 表雜湊值空間、s 表簽章值空間、t 表 hash 時間、ℓ 表 sign 時間、p 表 buffer size。分別說明如下:

Communication overhead: 鍊結性認證方法 BAFV 除了本文內容外,每個封包區塊都附加一個雜湊值 h，而傳遞給接收端之前，並將第一個封包區塊之雜湊值進行簽章，故其所需要額外使用的空間值為 s+nh，而 DFA、augmented chain、butterfly 則於每個封包區塊上附加二個雜湊值，故其額外使用空間值為 s+2nh，HCPI 也是於每個封包區塊附加二個雜湊值，但是起始封包區塊認證為使用單一封包認證的雜湊值進行認證,故其額外使用空間值為 2nh。

Verification/容錯 probability: 假若封包區塊傳送無遺失或錯誤時，所有的認證方法，於接收端時都可以繼續且完成內文認證,但是若有發生封包遺失或錯誤時,BAFV 無法繼續向下進行鍊結性認證，而 DFA、augmented chain、butterfly 等方法則需要考慮錯誤封包所發生的地方，因為此些方法都有二個認證碼可以進行認證，且錯誤封包通常為鄰近封包錯誤，對於 augmented chain、butterfly 方法中第二個認證碼存在於數個封包之外，此時則可以透過此些正確的認證碼進行認證，而單一封包區塊認證法亦可以繼續向下進行鍊結性認證，所以其容錯率為 100%。

Computation overhead: BAFV 方法在多媒體內容傳遞至接收端之前，先於發送端進行每個封包區塊的雜湊運算，最後進行簽章運算，故其簽章及雜湊運算時間為 ℓ +nt，而 DFA、augmented chain、butterfly 只是多放置雜湊值，並沒有多進行一次雜湊運算，故其簽章及雜湊運算時間亦為 ℓ

+nt，然而 HCPI 每個封包區塊則進行二次雜湊運算，一次為個別封包區塊的雜湊運算，一次為鍊結性封包區塊雜湊運算，所以 HCPI 的運算時間為 2nt。

Sender delay / receiver delay: BAFV、DFA、butterfly 等方法的 Sender delay 與 receiver delay，同為 n 及 1，因為這些方法是由最後的封包開始進行雜湊簽章，所以需要全數封包雜湊運算結束後，才進行簽章運算及傳遞，而認證時接到簽章封包後則可以開始進行認證，故其 Sender delay 與 receiver delay，為 n / 1；而 augmented chain 方法則由小單元封包區塊進行雜湊運算，但需要等到 global links 之封包運算完才可傳遞，所以 Sender delay 為 p，接收端要接收到最後一封包為簽章封包後，才可以進行認證，故其 receiver delay 為 n；HCPI 方法因為仍然有使用鍊結性認證方法 BAFV，需要練結性雜湊運算完所有封包區塊後才可以傳遞，而接收端收到第一個封包區塊則開始進行認證，故其 Sender delay 與 receiver delay，亦為 n / 1。

4.2 討論

安全性:

HCPI 認證法是以 BAFV 鍊結性認證為基礎再輔以封包獨立認證，因此其安全性可由 BAFV 鍊結性認證與封包獨立認證分別來探討。

a. 鍊結性認證:

相似於其它鍊結性認證法，HCPI 認證法使用安全度高的湊雜函數，如 MD5 或 SHA-1 產生鍊結封包的驗證值，因此其安全性與所使用的的湊雜函數相同。

b. 封包獨立認證:

因封包驗證碼亦是由封包編號密碼與內文經 MD5 或 SHA-1 湊雜運算產生，而封包編號密碼又需由樹狀結構轉換與密碼簿產生，因此要替換整個封包而不被察覺，必須有內文以及封包編號密碼，假設內文可由影像或聲音回推，封包編號密碼必須有正確的密碼簿及數狀結構秘密參數，如分支數，由於封包總數相當大，因此樹狀結構之階數與分支數可有相當多種的選擇，要正確的反推機率很小，而編碼簿是安全的傳送，因此封包獨立認證具有雙層的安全保護。

時間、空間:

相較於 augmented chain 或 butterfly 二方法在封包簽章之前需先對封包進行特定的鍊結排列，單一封包認證法在未開使各個封包簽章之前，亦需要將封包的排序編號轉為樹狀編號再編碼，外加一次的編碼簿傳遞，而數狀編碼演算法相當簡單，因此所需的時間不多，另外編編碼簿之大小則依數狀結構分支度而定，分支度愈大(樹高愈低)則需要的碼簿之大小愈大。相較於傳統傳遞雜湊表方法，若 1000 個封包區塊認證上，傳統傳遞雜湊表大小為 1000h 雜湊空間，而單一封包認證法之碼簿之大小則僅是 2h，因此其空間與時間的需求亦不高，值得一提的是，分支度過小，將大大影響透過編碼簿織編碼之安全性。

格式相容性:

單一封包認證法所使用的關鍵值為封包編號，亦即可透過編號進行封包本身的認證，因此不受限於任何的影音或檔案格式，更不需要有新的檔案格式或影音格式就需要配合一個適合的認證方法，相容性大。

5. 結論

傳統的單一封包獨立認證會衍生封包整個被置換而無法察覺的問題，而鍊結性認證雖可解決其缺失，卻存有因其中某一封包

的錯誤而無法繼續認證之容錯問題。

認證方法 BAFV 的出現，降低了認證上整體時間與空間的使用需求，但是由於 BAFV 為一鍊結性的認證方法，不具有認證的容錯性與彈性；而非鍊結性之封包獨立認證法雖可解決鍊結性認證法在容錯性與彈性之問題，卻存在著整個封包被換掉而無法察覺之缺失。我們在文中所提出的鍊結獨立混合性多媒體認證法可同時解決上述二種方法所存在的問題，並兼具其優點，既可將認證碼藏存入封包中以降低空間的使用需求，亦可察覺封包整塊被換置的情況以抵擋 VQ 攻擊，且具有 100%可略過錯誤或遺失封包的容錯率。更甚者，本方法以封包編號為封包驗證碼之關鍵值，因此也具有影音或檔案格式相容性。

參考文獻

[1] A. Perrig, R. Canetti, J. Tygar, and D. Song, “Efficient authentication and signing of multicast streams over lossy channels”, in Proc. IEEE Symp. Security and Privacy, pp. 56–73, 2000.

[2] D. Mukherjee, H. Wang, A. Said, S. Liu, “Format independent encryption of generalized scalable bit-streams enabling arbitrary secure adaptations”, Proc. IEEE Int. Conf Ac., Speech and Sig. Proc., Philadelphia, March 2005.

[3] D.Mukherjee, ”FORMAT-INDEPEN DENT AUTHENTICATION OF ARBITRARY SCALABLE BIT-STREAMS USING ONE-WAY ACCUMULATORS”, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Vol. 2, pp. II-829 - II-832, Apr. 2007.

[4] H. H. Yu, “A loss resilient and scalable streaming media authentication scheme”, Consumer Communications and Networking Conference, CCNC, IEEE, pp.60 - 64, Jan. 2005.

[5] H. H. Yu, “Scalable Multimedia Authentication”, Proc. Joint Conf. on Info., Comm. and Signal Proc., 2003 and 4th Pacific Rim Conf. on Multimedia (ICICS-PCM 2003), pp. 443 – 447, Singapore, Dec. 15-18, 2003.

[6] H. Yu, “Scalable streaming media authentication”, Proc. of IEEE ICC2004, Paris, France, Jun. , 2004.

[7] P. Golle and N. Modadugu, “Authenticating streamed data in the presence of random packet loss”, in ISOC Network and Distributed System Security Symp., pp. 13–22, 2001.

[8] R. Gennaro and P. Rohatgi, “How to sign digital streams”, in Advances in Cryptology—CRYPTO ’97, pp. 180–197, 1997.

[9] Z. Zhang, Q. Sun, and W.-C. Wong, “A proposal of butterfly-graph based stream authentication over lossy networks”, in Proc. IEEE Int. Conf. Multimedia & Expo, The Netherlands, Jul. 2005.

[10] Zhang Zhishou , Q. Sun ,Wong Wai-Choong, J. Apostolopoulos , S. Wee, “An Optimized Content-Aware Authentication Scheme for Streaming JPEG-2000 Images Over Lossy

Networks", IEEE Transactions on Multimedia, Vol. 9, Issue 2, pp. 320 – 331, Feb. 2007.

[11] MD5, available at http://en.wikipedia.org/wiki/MD5

[12] SHA-1, available at http://en.wikipedia.org/wiki/SHA-1