適用於 IoT 之輕量化鑑別機制

蔡國裕
華夏科技大學資訊管理系
kytsai@cc.hwh.edu.tw

摘要

隨著資訊通訊技術之發展，物聯網 (Internet of Things，簡稱 IoT)之應用能多方面影響我們的日常生活，甚且可以改變商業模式。然而，在某些應用中，諸如健康照護 (healthcare)，IoT 之隱私 (privacy)、安全 (security)及鑑別 (authentication)等相關議題為 IoT 發展之關鍵因素。本論文提出一個適用於 IoT 之輕量化鑑別機制，我們採用動態雜湊鏈 (dynamic hashing chain)設計節點 (node)間之鑑別方法。我們所提出之方法不僅可以達到低計算成本 (low computation costs)，亦可以達相關安全需求。

關鍵詞：物聯網、動態雜湊鏈、輕量化、鑑別

壹、前言

在物聯網 (Internet of Things，簡稱 IoT)世界中，經由各種不同感測裝置與通訊協定，可以進行資訊傳送與分享，可用於定位、監控及追蹤等不同應用。IoT 一詞首先由 Ashton [1]所提出，但是 IoT 於近幾年才從概念轉變成有實體產品，目前世界先進國家無不將發展 IoT 列為國家重大政策。然而，根據國際電信聯盟 (International Telecommunication Union，簡稱 ITU)於 2005 年所提出之報告指出[2]，要說服使用者採用新興技術之最重要的挑戰為資料與隱私保護。有關 IoT 安全架構之研究[3][4][8]相繼被提出。

在 2012 年，Wen 等人[8]提出適用於 IoT 之動態可變的加密安全憑證。在 Wen 等人之方法中，提出金鑰矩陣概念，通訊中之節點共享相同的金鑰矩陣，並透過隨機選取其中之金鑰，在鑑別協定中，不同回合所使用之加密金鑰皆為不同。然而，IoT 中可能存在低資源 (low resource)之節點，其計算能力有限，不適合執行傳統對稱式加密技術[10]。因此，在本論文中，我們提出兩個輕量化鑑別方法。在第一個方法中，我們使用 Wen 等人 [8]所提出之金鑰產生方式，但在加密技術上，我們採用計算成本較低之互斥或運算 (exclusive-or operation)取代 Wen 等人所採用的對稱式加密技術 (symmetric encryption technique)。在第二個方法中，我們採用動態雜湊鏈 (dynamic hashing chain)方法產生每回合所需之加密金鑰，由於單向雜湊函數 (one-way hash function)之輸入值包含時間戳記 (time-stamp)，所以每回合之加密金鑰皆為不同。再者，我們在兩個方法中皆採用金鑰雜湊訊息鑑別碼 (keyed-hash message authentication code，簡稱 HMAC)[6]確保

鑑別過程中所傳送訊息之完整性。即，當攻擊者欲篡改節點所傳送之訊息時，由於攻擊者沒有加密金鑰，無法產生相對之 HMAC，所以攻擊者無法成功篡改訊息。

本論文其他架構如下，第貳章回顧 Wen 等人 [8]所提出之方法，包括動態可變之加密設計概念與鑑別程序；第參章提出兩個輕量化鑑別方法，第一個方法採用互斥或運算作為加密運算，以降低計算成本，第二個方法採用雜湊鏈方式產生加密金鑰，以減少儲存成本；第肆章分析我們所提出之方法的安全性，包括節點之匿名性、訊息之機密性、訊息不可偽造、節點不可假冒及抵抗重送攻擊；最後一章為結論。

貳、Wen 等人所提出之適用於 IoT 之動態可變的加密安全憑證

在第貳章中，我們回顧 Wen 等人[8]於 2012 年所提出之動態可變的加密安全憑證，動態可變加密安全憑證為一種基於請求-應答機制之可變的鑑別協定。在 2.1 節中我們回顧動態可變之加密設計概念，主要應用金鑰矩陣概念。2.2 節則是介紹動態可變之加密安全憑證的鑑別程序，其所使用之符號定義如下表所示。

表一：符號定義表

符號	描述
A, B	A, B 為通訊節點 (communication node)。
ID_a, ID_b	ID_a 與 ID_b 分別代表節點 A 之身分識別碼 (identity number)與節點 B 的身分識別碼。
conReq	*conReq* 代表連線需求。
$coorXY_i$	$coorXY_i$ 為金鑰矩陣中之 x 軸座標值與 y 軸座標值，其中 i 代表回合數。
TS_i	TS_i 代表時間戳記，其中 i 代表回合數。
K_1	K_i 為通行碼，作為對稱式加密演算法之金鑰。
$E_{K_i}(Msg_1 \| Msg_2 \| \ldots \| Msg_n)$	$E_{K_i}(Msg_1 \| Msg_2 \| \ldots \| Msg_n)$為對稱式加密演算法，其中 K_i 為通訊節點之共享密的通行碼，作為加密金鑰； Msg_i 為欲加密之訊息；$\|$為串連(concatenation)運算子。
msgCon	*msgCon* 為訊息常數。

2.1 動態可變之加密設計概念

欲進行通訊之節點需要共享相同金鑰矩陣，此金鑰矩陣為八列、八行之矩陣 (簡稱八乘八矩陣)，矩陣之元素大小為 8 個位元組，整個金鑰矩陣之儲存空間為 256 個位元組，如下圖一所示。通訊雙方根據金鑰矩陣之 *x* 軸(水平方向)與 *y* 軸(垂直方向)可以得到雙方共享之通行碼，此通行碼作為加密金鑰之用。在進行鑑別過程中，雙方僅需傳送金鑰矩陣之座標軸，而不是金鑰本身。即使是攻擊者攔截金鑰矩陣座標軸，亦無法獲得金鑰。此外，密碼長度之變化從最小 4 個位元組至最大 256 個位元組，其理論值可達到 $1.26*10^{89}$ 組通行碼，Wen 等人宣稱可達到真正一次一組加密，即每次加密之通行碼為不同。

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	35tq	15fg	Hg87	l23f	Na01	bzla	akjl	37ja
2	ghjk	omkt	12hp	d04h	39nz	AmjK	qfda	0k2v
3	3tbu	36b9	Fh01	Jk76	60az	1ghw	jzqp	qjkp
4	plmh	Fh8k	lj09	xaj0	bL09	jkbk	7aln	mb0z
5	Fn04	M7a0	9hja	z1k0	abaj	qjoa	adl9	ajlk
6	mk08	ed5l	L14g	ajkj	LzY5	hzq0	0qbo	F9Z1
7	F9zq	N081	sf34	K0JN	aznq	hjql	qjk0	76na
8	3ftl	Ly08	MJq1	bz1k	97ma	a9l7	HalU	aI3b

圖一：金鑰矩陣

2.2 動態可變之加密安全憑證的鑑別程序

假設有 *A*、*B* 兩節點進行通訊，其中節點 *A* 為客戶端節點 (client node)、節點 *B* 為伺服端節點 (server node)。兩節點共享之金鑰矩陣如圖一所示，通訊節點可以根據座標軸取得共享之金鑰，以作為加密金鑰。節點 *A* 與節點 *B* 共同執行下列步驟以進行鑑別 (如圖二所示)，詳述如下：

步驟 1： 節點 *A* 選擇金鑰矩陣之座標軸 $coorXY_1$，並根據金鑰矩陣取得本回合用於加密的通行碼 K_1。

步驟 2： 節點 *A* 使用 K_1 加密身分識別碼 ID_a、連線需求 *conReq* 及時間戳記 TS_1：

$$C_1 = E_{K_1}(ID_a \| conReq \| TS_1)$$。

步驟 3： 節點 *A* 將 $coorXY_1$ 與加密訊息 C_1 傳送給節點 *B*。

步驟 4： 當接收 $coorXY_1$ 與 C_1 之後，節點 *B* 根據 $coorXY_1$ 取得金鑰矩陣中之 K_1，進而用以解密 C_1。

步驟 5： 解密 C_1 之後，節點 *B* 可以取得 ID_a、*conReq* 及 TS_1，節點 *B* 驗證 TS_1 以確

保加密訊息 C_1 是否為在有效時間內所傳送。當此訊息是在有效時間內所接收，則節點 B 確認節點 A 之身分為正確，並繼續執行下一個步驟；否則，節點 B 拒絕所接收之連線需求，並停止執行。

步驟 6： 節點 B 選擇金鑰矩陣之座標軸 $coorXY_2$，並根據金鑰矩陣取得回應訊息所需之加密的通行碼 K_2。

步驟 7： 節點 B 使用 K_2 加密身分識別碼 ID_b、所選擇之金鑰矩陣之座標軸 $coorXY_3$、所接收之時間戳記 TS_1 及目前時間戳記 TS_2：

$$C_2= E_{K_2}(ID_b\| coorXY_3\| TS_1\| TS_2)。$$

步驟 8： 節點 B 將 $coorXY_2$ 與加密訊息 C_2 回傳到 A。

步驟 9： 對加密訊息 C_2 進行解密之後，節點 A 可以取得 ID_b、$coorXY_3$、TS_1 及 TS_2，節點 A 驗證 TS_2 以確保加密訊息 C_2 是否為在有效時間內所傳送。當 C_2 是在有效時間內所接收之訊息，則節點 A 確認節點 B 之身分為合法，並繼續執行下一個步驟；否則，節點 A 拒絕所接收之訊息。

步驟 10： 節點 A 根據座標軸 $coorXY_3$ 取得通行碼 K_3，並對訊息常數 $msgCon$、ID_a、所接收到之 B 所傳送的時間戳記 TS_2 及目前時間戳記 TS_3：

$$C_3= E_{K_3}(msgCon\| ID_a\| TS_2\| TS_3)。$$

節點 *A*		**節點 *B***
1. 選擇 $coorXY_1$，並取得 K_1		
2. 計算 $C_1= E_{K_1}(ID_a\| conReq\| TS_1)$		
	3. $coorXY_1$ 與 C_1 →	
		4. 根據 $coorXY_1$ 取得 K_1
		5. 解密 C_1，並驗證 TS_1 以確定 A 之身分
		6. 選擇 $coorXY_2$，並取得 K_2
		7. 計算 $C_2= E_{K_2}(ID_b\| coorXY_3\| TS_1\| TS_2)$
	← 8. $coorXY_2$ 與 C_2	
9. 解密 C_2，並驗證 TS_2 以確定 A 之身分		
10.選擇 $coorXY_3$，並計算 $C_3= E_{K_3}(msgCon\| ID_a\| TS_2\| TS_3)$		

圖二：鑑別程序示意圖

完成上列步驟後，代表節點A與節點B已經完成相互鑑別 (mutual authentication)，並在彼此間建立通道。然而，IoT之部份節點可能屬於低資源裝置，即節點計算能力有限，

不適合執行傳統對稱式加密技術 [10]。

參、我們所提出之基於動態雜湊鏈的IoT鑑別機制

本章介紹我們所提出之兩種輕量化鑑別方法，在 3.1 節中，我們仍採用 Wen 等人[8]所提出之金鑰矩陣概念，而在鑑別程序中，我們採用互斥或運算取代對稱式加密技術，其中因為互斥或運算之金鑰長度必須與訊息長度相等，所以每一回合中必須將金鑰串連數倍以等同訊息長度。再者，採用金鑰雜湊訊息鑑別碼[6]確保傳送訊息之完整性。在 3.2 節，我們進一步採用雜湊鏈方式取代金鑰矩陣，除降低儲存成本之外，仍可以達到每次加密之金鑰皆為不同。我們所使用之符號除部份與表一相同外，額外符號定義如表二所示。

表二：我們所提出之機制的符號定義表

符號	描述
$KH(K_i, Msg_1 \| Msg_2 \| \ldots \| Msg_n)$	$KH(K_i, Msg_1 \| Msg_2 \| \ldots \| Msg_n)$為單向雜湊函數，其輸入值為對稱式金鑰 K_i 與訊息 $Msg_1, Msg_2, \ldots, Msg_n$，輸出值為金鑰雜湊訊息鑑別碼 [6]。
$iniKey$	$iniKey$ 為 A 與 B 雙方共享之初始祕密值。
$H(iniKey \| TS_i)$	$H(iniKey \| TS_i)$代表單向雜湊函數 [5]。
$H_{i,j}(iniKey \| TS_i)$	$H_{i,j}(iniKey \| TS_i)$為單向雜湊鏈，其中 $H_{i,1}(iniKey \| TS_i) = H(iniKey \| TS_i)$、$H_{i,j+1}(iniKey \| TS_i) = H(H_{i,j}(iniKey \| TS_i))$、$i$ 代表回合數、$1 \leqq j \leqq m\text{-}1$，且 m 代表欲產生之金鑰最大數。
$K_{i,j}$	$K_{i,j}$ 加密金鑰，i 代表回合數，而 $1 \leqq j \leqq m$，其中 m 代表欲產生之金鑰最大數。

3.1 我們所提出之第一個方法

在第一個方法中，我們採用互斥或運算來取代對稱式加密技術，並以訊息鑑別碼確保訊息之完整性，即確認訊息沒有遭受篡改。當 A、B 兩節點欲進行通訊時，必須共同執行下列步驟以進行鑑別 (如圖三所示)，詳述如下：

步驟 1： 節點 A 選擇金鑰矩陣之座標軸 $coorXY_1$，並根據金鑰矩陣取得本回合用於加密的通行碼 K_1。

步驟 2： 節點 A 計算身分識別碼 ID_a、連線需求 $conReq$ 及時間戳記 TS_1 之訊息鑑別碼 MAC_1：

$$MAC_1= KH(K_1, ID_a|| conReq|| TS_1)。$$

步驟 3： 節點 A 將 K_1 串連數倍以等同訊息長度，用以加密 ID_a、$conReq$、TS_1 及 MAC_1：

$$C_1= (K_1|| K_1|| \ldots|| K_1)\oplus(ID_a|| conReq|| TS_1|| MAC_1)。$$

步驟 4： 節點 A 將 $coorXY_1$ 與加密訊息 C_1 傳送給節點 B。

步驟 5： 當接收 $coorXY_1$ 與 C_1 之後，節點 B 根據 $coorXY_1$ 取得金鑰矩陣中之 K_1，進而用以解密 C_1。

步驟 6： 解密 C_1 之後，節點 B 可以取得 ID_a、$conReq$、TS_1 及 MAC_1，節點 B 驗證 TS_1 與 MAC_1 以確保加密訊息 C_1 是否為在有效時間內所傳送，且訊息沒有遭受任何篡改。當驗證正確時，則代表節點 B 確認節點 A 之身分為正確，並繼續執行下一個步驟；否則，節點 B 拒絕所接收之連線需求，並停止執行。

步驟 7： 節點 B 選擇金鑰矩陣之座標軸 $coorXY_2$，並根據金鑰矩陣取得回應訊息所需之加密的通行碼 K_2。

步驟 8： 節點 B 計算身分識別碼 ID_b、所選擇之金鑰矩陣之座標軸 $coorXY_3$、所接收之時間戳記 TS_1 及目前時間戳記 TS_2 之訊息鑑別碼 MAC_2：

$$MAC_2= KH(K_2, ID_b|| coorXY_3|| TS_1|| TS_2)。$$

步驟 9： 節點 B 使用 K_2 串連數倍以等同訊息長度，用以加密 ID_b、$coorXY_3$、TS_1、TS_2 及 MAC_2：

$$C_2= (K_2|| K_2|| \ldots|| K_2)\oplus(ID_b|| coorXY_3|| TS_1|| TS_2|| MAC_2)。$$

步驟 10：節點 B 將 $coorXY_2$ 與加密訊息 C_2 回傳到 A。

步驟 11：對加密訊息 C_2 進行解密之後，節點 A 可以取得 ID_b、$coorXY_3$、TS_1、TS_2 及 MAC_2，節點 A 驗證 TS_2 與 MAC_2 以確保加密訊息 C_2 是否為在有效時間內所傳送，且訊息沒有遭受任何篡改。當 C_2 是在有效時間內所接收之訊息，則節點 A 確認節點 B 之身分為合法，並繼續執行下一個步驟；否則，節點 A 拒絕所接收之訊息。

步驟 12：節點 A 根據座標軸 $coorXY_3$ 取得通行碼 K_3，並計算訊息常數 $msgCon$、ID_a、所接收之時間戳記 TS_2、目前之時間戳記 TS_3 之訊息鑑別碼 MAC_3：

$$MAC_3= KH(K_3, msgCon|| ID_a|| TS_2|| TS_3)。$$

步驟 13：節點 A 使用 K_3 串連數倍以等同訊息長度，並對 $msgCon$、ID_a、TS_2、TS_3 及 MAC_3 進行加密：

$$C_3= (K_3|| K_3|| \ldots|| K_3)\oplus(msgCon|| ID_a|| TS_2|| TS_3|| MAC_3)。$$

節點 *A*		節點 *B*
1. 選擇 $coorXY_1$，並取得 K_1		
2. 計算 $MAC_1= KH(K_1, ID_a\|\| conReq\|\| TS_1)$		
3. 計算 $C_1 = (K_1\|\| K_1\|\| \ldots\|\| K_1)\oplus(ID_a\|\| conReq\|\| TS_1\|\| MAC_1)$		
	4. $coorXY_1$ 與 C_1 →	
		5. 根據 $coorXY_1$ 取得 K_1
		6. 解密 C_1，並驗證 TS_1 與 MAC_1，以確定 *A* 之身分
		7. 選擇 $coorXY_2$，並取得 K_2
		8. 計算 $MAC_2= KH(K_2, ID_b\|\| coorXY_3\|\| TS_1\|\| TS_2)$
		9. 計算 $C_2= (K_2\|\| K_2\|\| \ldots\|\| K_2)\oplus(ID_b\|\| coorXY_3\|\| TS_1\|\| TS_2\|\| MAC_2)$
	← 10. $coorXY_2$ 與 C_2	
11.解密 C_2，並驗證 TS_2 與 MAC_2，以確定 *A* 之身分		
12.選擇 $coorXY_3$，並計算 $C_3= (K_3\|\| K_3\|\| \ldots\|\| K_3)\oplus(msgCon\|\| ID_a\|\| TS_2\|\| TS_3\|\| MAC_3)$		

圖三：第一個方法之鑑別程序示意圖

3.2 我們所提出之第二個方法

在第二個方法中，我們採用雜湊鏈產生所需之金鑰數量，以取代 Wen 等人 [9]所提出之金鑰矩陣概念。當 *A*、*B* 兩節點欲進行通訊時，必須共同執行下列步驟以進行鑑別 (如圖四所示)，詳述如下：

步驟 1： 節點 *A* 計算第一回合所需之金鑰數 $K_{1,1}= H_{1,2}(iniKey\|\| TS_1)= H(iniKey\|\| TS_1)$, $K_{1,2}= H_{1,2}(iniKey\|\| TS_1)$, …, $K_{1,m}= H_{1,m}(iniKey\|\| TS_1)$，其中 TS_1 為時間戳記、*m* 代表欲產生之金鑰最大數，且 $H_{1,j+1}(iniKey\|\| TS_1) = H(H_{1,j}(iniKey\|\| TS_1))$，$1 \leqq j \leqq m\text{-}1$。

步驟 2： 節點 *A* 計算身分識別碼 ID_a、連線需求 *conReq* 及 TS_1 之訊息鑑別碼 MAC_1：

節點 *A*		節點 *B*
1. 計算 m 把金鑰 $K_{1,1}= H(iniKey\|\| TS_1)$, $K_{1,2}= H_{1,2}(iniKey\|\| TS_1)$, …, $K_{1,m}= H_{1,m}(iniKey\|\| TS_1)$		
2. 計算 $MAC_1= KH(K_1, ID_a\|\| conReq\|\| TS_1)$		
3. 計算 $C_1= (K_{1,1}\|\| K_{1,2}\|\| \ldots\|\| K_{1,m})\oplus (ID_a\|\| conReq\|\| MAC_1)$	4. TS_1 與 C_1 →	
		5. 根據 TS_1 分別計算金鑰 $K_{1,1}= H(iniKey\|\| TS_1)$, $K_{1,2}= H_{1,2}(iniKey\|\| TS_1)$, …, $K_{1,m}= H_{1,m}(iniKey\|\| TS_1)$
		6. 解密 C_1，並驗證 TS_1 與 MAC_1，以確定 A 之身分
		7. 計算 $K_{2,1}= H(iniKey\|\| TS_2)$, $K_{2,2}= H_{2,2}(iniKey\|\| TS_2)$, …, $K_{2,m'}= H_{2,m'}(iniKey\|\| TS_2)$
		8. 計算 $K_{3,1}= H(iniKey\|\| TS_1\|\| TS_2)$
		9. 計算 $MAC_2= KH(K_{2,1}, ID_b\|\| K_{3,1}\|\| TS_1\|\| TS_2)$
		10. 計算 $C_2= (K_{2,1}\|\| K_{2,2}\|\| \ldots\|\| K_{2,m'})\oplus (ID_b\|\| K_{3,1}\|\| MAC_2)$
	← 11. $coorXY_2$ 與 C_2	
12. 解密 C_2 取得 ID_b、$K_{3,1}$ 及 MAC_2，並根據 TS_1 與 TS_2 計算 $K_{3,1}$，驗證 TS_2 與 MAC_2，以確定 A 之身分		
13. 計算 $MAC_3= KH(K_3, msgCon\|\| ID_a\|\| TS_2\|\| TS_3)$		
14. 計算 $K_{3,2}= H(K_{3,1}), \ldots, K_{2,m''}= H(H_{2,m''-1}(iniKey\|\| TS_1\|\| TS_2\|\| TS_3))$		
15. 計算 $C_3= (K_3\|\| K_3\|\| \ldots\|\| K_3)\oplus (msgCon\|\| ID_a\|\| TS_2\|\| TS_3\|\| MAC_3)$		

圖四：第二個方法之鑑別程序示意圖

$MAC_1= KH(K_{1,1}, ID_a\|\| conReq\|\| TS_1)$。

步驟 3： 節點 A 加密 ID_a、$conReq$ 及 MAC_1：

$C_1 = (K_{1,1} \| K_{1,2} \| \ldots \| K_{1,m}) \oplus (ID_a \| conReq \| MAC_1)$。

步驟 4： 節點 A 將 TS_1 與加密訊息 C_1 傳送給節點 B。

步驟 5： 當接收 TS_1 與 C_1 之後，節點 B 根據 TS_1 分別計算金鑰 $K_{1,1} = H(iniKey \| TS_1)$, $K_{1,2} = H_{1,2}(iniKey \| TS_1)$, …, $K_{1,m} = H_{1,m}(iniKey \| TS_1)$。

步驟 6： 解密 C_1 之後，節點 B 可以取得 ID_a、$conReq$ 及 MAC_1，節點 B 驗證 TS_1 與 MAC_1 以確保加密訊息 C_1 是否為在有效時間內所傳送，且訊息沒有遭受任何篡改。當驗證正確時，則代表節點 B 確認節點 A 之身分為正確，並繼續執行下一個步驟；否則，節點 B 拒絕所接收之連線需求，並停止執行。

步驟 7： 節點 B 計算第二回合所需之金鑰數 $K_{2,1} = H(iniKey \| TS_2)$, $K_{2,2} = H_{2,2}(iniKey \| TS_2)$, …, $K_{2,m'} = H_{2,m'}(iniKey \| TS_2)$，其中 m' 代表欲產生之金鑰最大數。

步驟 8： 節點 B 計算第三回合所需之第一把金鑰 $K_{3,1} = H(iniKey \| TS_1 \| TS_2)$。

步驟 9： 節點 B 計算身分識別碼 ID_b、所接收之時間戳記 TS_1 及目前時間戳記 TS_2 之訊息鑑別碼 $MAC_2 = KH(K_{2,1}, ID_b \| K_{3,1} \| TS_1 \| TS_2)$。

步驟 10：節點 B 加密 ID_b、$K_{3,1}$ 及 MAC_2：

$C_2 = (K_{2,1} \| K_{2,2} \| \ldots \| K_{2,m'}) \oplus (ID_b \| K_{3,1} \| MAC_2)$。

步驟 11：節點 B 將 TS_2 與加密訊息 C_2 回傳到 A。

步驟 12：對加密訊息 C_2 進行解密之後，節點 A 可以取得 ID_b、$K_{3,1}$ 及 MAC_2，並根據 TS_1 與 TS_2 計算 $K_{3,1}$。節點 A 驗證 TS_2 與 MAC_2 以確保加密訊息 C_2 是否為在有效時間內所傳送，且訊息沒有遭受任何篡改。當 C_2 是在有效時間內所接收之訊息，則節點 A 確認節點 B 之身分為合法，並繼續執行下一個步驟；否則，節點 A 拒絕所接收之訊息。

步驟 13：節點 A 計算訊息常數 $msgCon$、ID_a、所接收之時間戳記 TS_2、目前之時間戳記 TS_3 之訊息鑑別碼 $MAC_3 = KH(K_3, msgCon \| ID_a \| TS_2 \| TS_3)$。

步驟 14：節點 A 計算 $K_{3,2} = H(K_{3,1})$, …, $K_{2,m''} = H(H_{2,m''-1}(iniKey \| TS_1 \| TS_2 \| TS_3))$，其中 m'' 代表欲產生之金鑰最大數量。

步驟 15：節點 A 加密 $msgCon$、ID_a、TS_2、TS_3 及 MAC_3：

$C_3 = (K_{3,1} \| K_{3,2} \| \ldots \| K_{3,m''}) \oplus (msgCon \| ID_a \| MAC_3)$。

肆、安全分析

基於單向雜湊函數之安全假設 [7]，我們將證明所提出之方法的安全性，單向雜湊函數之描述如下。

[單向雜湊函數]令 H 為單向雜湊函數，其輸入值為任意長度之訊息 m，輸出值為雜湊值 $H(m)$。具有下列三項性質：(1) 給定雜湊值 $H(m)$，推導出 m 為計算上不可行；(2) 找出

兩個不同訊息 m 與 m' 滿足 $H(m)= H(m')$ 為計算上不可行；(3) 存在一個有效率地計算 $H(m)$ 之演算法。

以下分析我們所提出之方法滿足節點之匿名性、訊息之機密性、訊息不可偽造、節點不可假冒及抵抗重送攻擊：

(1) 節點之匿名性：除通訊節點外，其餘第三者無法得知節點之身分識別碼。
   (i) 在第一個方法中，假設攻擊者攔截傳送訊息取得 C_1、C_2 或 C_3，由於攻擊者無法取得金鑰，則攻擊者嘗試猜測金鑰值，然而攻擊者成功機率為 $1/2^{|K_i|}$，其中 $|K_i|$ 為金鑰長度。因此，若要提高安全強度，則選用較長之加密金鑰。
   (ii) 在第二個方法中，假設攻擊者亦進行同樣攻擊，攻擊者成功機率為 $1/2^{|H|}$，其中 $|H|$ 為單向雜湊函數之雜湊值長度。在第二個方法中，要要提高安全強度，則選用輸出值之位元長度較長之單向雜湊函數。

(2) 訊息之機密性：除通訊節點外，其餘第三者無法得知傳送訊息之內容。本安全分析與節點之匿名性相同，其安全強度與金鑰長度有關。

(3) 訊息不可偽造：除合法節點外，其餘第三者無法偽造成有效訊息。
   (i) 在第一個方法中，假設攻擊者欲偽造第一回合之有效訊息，由於攻擊者無法取得金鑰，則攻擊者嘗試隨機產生一把金鑰 K'_1 與座標軸 $coorXY_1$，並計算 $MAC_1= KH(K'_1, ID_a\| conReq\| TS_1)$。由於單向雜湊函數之假設，攻擊者欲找到 K'_1 與 K_1(通訊節真正共享之金鑰)滿足 $KH(K'_1, ID_a\| conReq\| TS_1)= KH(K_1, ID_a\| conReq\| TS_1)$ 為計算上不可行。若攻擊者欲偽造第二回合之合法訊息，安全分析與前述相同，安全性皆基於單向雜湊函數之假設。
   (ii) 在第二個方法中，假設攻擊者欲偽造第一回合之有效訊息，由於攻擊者無法取得金鑰，則攻擊者嘗試隨機選擇 $iniKey'$ 與計算金鑰 $K'_{1,1}= H(iniKey'\| TS_1)$，並進而計算 $MAC_1= KH(K_{1,1}, ID_a\| conReq\| TS_1)$。由於單向雜湊函數之假設，攻擊者欲找到 K'_1 與 K_1(通訊節真正共享之金鑰)滿足 $H(iniKey'\| TS_1)= H(iniKey\| TS_1)$ 為計算上不可行。若攻擊者欲偽造第二回合之合法訊息，安全分析與前述相同。

(4) 節點不可假冒：任何第三者皆無法成功假冒合法節點。若攻擊者欲假冒合法節點，則必須計算出有效訊息鑑別碼，分析與訊息不可偽造相同。

(5) 抵抗重送攻擊：假設攻擊者攔截節點間所傳送之訊息 C_1 或 C_2，並嘗試重送訊息。然而，在我們所提出兩個方法中，皆包含時間戳記，可以抵抗重送攻擊。若攻擊者欲替換以新時間戳記替換後重送訊息，則攻擊者將面臨如(3)之分析。

伍、結論

我們提出兩個適用於 IoT 之輕量化鑑別方法，兩個方法皆使用互斥或運算進行加密，可減少計算成本。此外，第二個方法則是採雜湊鏈方式動態產生金鑰，減少儲存成本。我們所提出之方法皆可以達到節點之匿名性、訊息之機密性、訊息不可偽造、節點不可假冒及抵抗重送攻擊。

[誌謝]
本研究部份接受科技部研究計畫經費補助，計畫編號：103-2221-E-146-005-MY2 、103-2221-E-011-090-MY2及104-2119-M-011-003。

參考文獻

[1] K. Ashton, "That 'Internet of things' thing," *RFID Journal*, 22 June 2009.
[2] International Telecommunication Union, "The Internet of Things," *ITU Internet Reports*, 2005.
[3] Z. Li, "The research of internet of things security issues," *Network and Computer Security*, Vol. 10, pp. 57-59, 2011.
[4] G. Li, Y. Bo and S. Yan, "Study on secure system architecture of IOT," *Information Security and Communications Privacy*, Vol. 12, pp. 73-75, 2010.
[5] National Institute of Standards and Technology, "Secure Hash Standard (SHS)," *Federal Information Processing Standards Publication 180-4 (FIPS PUB180-4)*, National Institute of Standards and Technology, 2015.
[6] National Institute of Standards and Technology, "The keyed-hash message authentication code (HMAC)," *Federal Information Processing Standards Publication 198-1 (FIPS PUB198-1), National Institute of Standards and Technology*, 2008.
[7] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, JohnWiley & Sons, New York, NY, USA, 2nd edition, 1996.
[8] Q. Wen, X. Dong and R. Zhang, "Application of dynamic variable cipher security certificate in Internet of things," *IEEE 2nd International Conference on Cloud Computing and Intelligent Systems (CCIS)*, Vol. 03, pp.1778-1783. ,2012
[9] C. Wu, "A preliminary investigation on the security architecture of the internet of things," Bulletin of the Chinese Academy of Sciences, Vol. 25, No. 4, pp. 411-419, 2010.
[10] 游佩芬，"漫談物聯網在醫療照護應用"， *IEK 產業情報網*，工業技術研究院，2011。

[作者簡介]

蔡國裕博士分別於 2001 年與 2009 年取得臺灣科技大學資訊管理系碩士學位與博士位。2009 年 9 月至 2012 年 7 月，蔡博士於臺灣科技大學資通安全研究與教學中心服研發替代役 (博士後研究員)。研發替代役結束後，蔡博士進入華夏科技大學資訊管理擔任助理教授。蔡博士為中華民國資訊安全學會永久會員，研究興趣包括密碼學、資訊安全、物聯網應用安全及雲端運算應用安全等。