改善 LEAP+中
全域金鑰更新之效能

張智超
長榮大學資訊管理學系
Email:terence@mail.cjcu.edu.tw

陳穎豪
長榮大學資訊管理學系
Email:cia740415@hotmail.com

摘要

內部攻擊在無線感測器網路中察覺不易且排除困難，在許多方法中，較有效率且普遍可行的做法，是利用全域金鑰來完成排除非法成員的動作。在 LEAP+(Localized Encryption and Authentication Protocol+)中，在效能與硬體許可的情形下，使用了四種金鑰來達成無線感測器中多種通訊需求的安全需求，但是其中全域金鑰的更新程序會消耗大量的運算能量，因此攻擊者將可以藉由觸發大量的全域金鑰更新而迅速降低 LEAP+的生存時間(lifetime)。

本篇論文提出一個適當的解決方式，策略性的利用感測器記憶體內現有的鄰居成員資訊來提升 LEAP+全域金鑰的更新效能。另外，此法也能套用到其他使用全域金鑰的安全協定中。

關鍵字：LEAP+，Wireless Sensor Networks，Global key，Security。

一、緒論

1.1. 無線感測器網路中全域金鑰更新

無線感測器網路是由多個感測器組合成的無線網路，若是需要大量的使用感測器的環境，基於成本考量，這些感測器多半價廉且運算功能偏弱。在目前針對無線感測器網路安全性的研究，合理且可行的通訊協定必須配合感測器的能力來運作，為了安全性的考量加入了過多的額外運算是不可行的，因此多數的安全協定會在能量許可的範圍之內，global key 利用簡單的安全性概念來運作，所有成員共享一把金鑰，使用便利且節省能源的 global key 常被作為提升無限感測器網路安全性的工具。global key 主要作為基地台與整個網路通訊的加密，使用上相較於其他一對一的金鑰或是區域性的金鑰更加的節省能量。

一般無線感測器網路的安全性需求分為三個階段，部署前、部署中與部署後。在部署前每個感測器是一個獨立的個體，在沒有通訊功能的情況下敵人只能複製感測器或是偽裝成某個感測器加入，但這幾乎不可行。在部署時，多數的協定是利用感測器交互發出訊息來與附近的感測器首次溝通而有了訊息的交流，因為網路架構處於還在架設的階段，大部分的安全協定會假設部署會在一段相當短的時間內完成且敵人無法在時間內完成攻擊，因此部署期間敵人的攻擊不會成功，也有少數的安全協定會對這部分做出防範，像 LEAP+[1,2]限制部署的時間，等於限制了敵人可以攻擊的時間來降低被攻擊成功的機率。無限感測器網路主要與敵人的攻防戰是在網路部署後展開，敵人可以主動的攻擊無限感測器網路。在竄改，竊取等多樣化的攻擊其中最棘手的稱為綁架攻擊(compromised attack)，也就是敵人綁架了某個網路中的成員，並且握有這個成員手上一切的資訊。因此當基地台發現某個成員被綁架之後，不管是

使用何種金鑰的安全協定都必須盡快切斷被綁架成員與網路間的關係讓網路可以繼續安全運作。但在被綁架成員洩漏了所有的訊息後，光是切斷關係是不夠的，還需要更新所有可能已經被敵人知道加密工具，其中當然也包括了 global key。

二、文獻探討

2.1.無線感測器網路內全域金鑰應用

最早的全域金鑰概念，在安全的概念初成形時就已經有一個最單純的作法，團體內的所有成員共享一個秘密(secret)作為通訊的加解密工作，也就是對稱式加密金鑰系統。因為原理簡單，到了現在仍然是一個節省能源並且能夠給予足夠安全性的方法，在無線感測器網路這個領域中也不例外。在一個正常運作的無線感測網路中如果有安全性上的需求，就會套入適當的無線感測網路安全協定，像是 SPIN[3](Security Protocols for Sensor Networks)，TinySec[4]或是 SM[5](Security Manager)這些安全協定。而多數安全協定當中除了利用各式各樣的技巧加入安全性之外，多數的協定會使用 global key 的概念。

我們可以把無線感測網路分成三個部份來看，基地台(base station)，感測器(sensors)還有敵人(adversary)。基地台通常是整個網路的核心，所有相關的任務都是由基地台發佈，或是所有的資訊最後需要回傳到基地台，以戰爭的角度來看就是我們說的作戰控制中心。感測器是分布在整個開放空間中的衛兵，它們聽從基地台的指示，忠實而且精密的執行任何接收到的任務。敵人無時無刻想從這些感測器手中知道基地台發送給他們的任務，竊取、追蹤、竄改與監視，都是敵人常用的手法。為了防止敵人可以輕易的得知當前的任務，學者們第一個想到的對應辦法就是把基地台的任務資訊先加密後再發送到整個無線感測網路，這就是 global key 最開始的概念。

因為 global key 的作法相當實用，只要網路中所有成員都擁有一把相同的 global key，基地台在發佈任務訊息的時候只需要利用 global key 加密，整個網路就可以快速且有效率的了解任務內容，利用預先寫入(pre-load)的方式植入所有的感測器當中，但是當敵人的手法更加多樣的時候我們發現如果不適時的更新 global key，當敵人知道了這把唯一的 global key，這個安全系統就形同虛設。於是 global key 的更新方式成為現在研究 global key 的學者主要的目標，既然更新是必須的，如何更安全、更快速又更節省能量的更新成為重點。多數學者同意 global key 的更新除了隨著時間經過有規律的更新之外，另外一個更新的時機就是在 compromised node 被發現之後，藉由更新 global key 來排除這個非法成員。

2.2.現今全域金鑰發展

無線感測器體積小且運算能力較弱，但在科技發達的現在，感測器日新月異，甚至有些已經能夠進行公開金鑰一類的運算。Staddon,J 與 Hong,T 在 2008 年分別提出 Self-Healing 法[6]與 TGDH[7](Tree base Diffie-Hellman)。這兩個方法是早期無線感測器網路無法接受的做法，他們壓縮公開金鑰複雜的運算內容來讓方法可行，因為過於龐大的運算會導致網路的生存時間快速縮短，不過硬體的進步讓這些協定的可行性大增，複雜的運算相對的也提供較高的安全性，雖然公開金鑰匙的方法安全性相對於對稱式加密高出許多，但是對於公開金鑰過於複雜的運算，這兩個協定降低的運算成本還不夠讓一般價廉的感測器使用。

在 Staddon,J 的 Self-Healing 法

中，主要是利用協定本身的多項式運算來完成遺失封包的重新獲取，也就是說某個成員在某次廣播後並沒有收到該收到訊息，但是再下一次廣播收到後，這個成員便可以利用此訊息與上一次收到的訊息來重新獲得未收到的訊息。Li,L[8]修改此法，在原本的多項式上做調整，使新的方法可以更有效率的更新 global key，進而提升安全性。Gerlbayar, T[9]提出原 Self-Healing 法中有不需要的元素存在，他利用隨機函式來產生 global key，簡化 Self-Healing 法的運算過程。我們可以發現多樣的能力發展與簡化金鑰運算過程是 Self-Healing 法後續研究的重心，初期的 Self-Healing 法只能提供訊息的重新獲得能力，但在學者加以改良之後也獲得了一般無線感測器網路安全協定該有的安全性，剩下的問題是如何在運算能力較強的感測器普及前努力的簡化 Self-healing 的運算，將來此法的發展勢必朝此方向延伸。

在 Kym,Y 的 TGDH 法中，主要概念是將無線感測器網路中所有成員間的關係建立成一個樹(tree)的結構，如此一來方便利用我們已知所有樹的運算來協助無線感測器網路的運作，把基地台當作 root，從金鑰的分發到訊息的傳送，都是利用這個結構來達成，最初的 TGDH 是作為新成員加入與舊成員離開的驗證協定。但在後續的改良後也可以進行一般無線感測器網路安全協定的作業。Hong,T[10]改善 TGDH protocol，透過驗證 two-party 和 three-party 來完成金鑰的運算，進而完成自己的方法能做到 TGDH 無法完成的隱密金鑰驗證(implicit key authentication)，也比 TGDH 更有效率。Poornima,A.S.[11]改良 TGDH，除了減少通訊和運算的成本外，當有成員被綁架的時候使用 one way key chain 和簡單的 XOR 運算來進行防禦動作。從幾篇後續的研究中發現，TGDH 的未來發展與 Self-Healing 法相似，除了給予更多一般無線感測器網路安全協定的能力之外，剩下的就是簡化運算來提升效能。

雖然硬體不斷的進步，但是"可行"與"普遍可行"有著顯著的差別。因此有另外一部分的學者並沒有放棄早期的無線感測器安全協定全域金鑰的研究，像是 Zhu,S 提出的 LEAP+。Localized Encryption and Authentication Protocol+(LEAP+)是一個金鑰管理協定(key management protocol)，藉由提供高達四種的金鑰型態來應付無線感測網路多樣的通訊。在 LEAP+當中，金鑰的建立方式就像它的使用方式一樣是一步接著一步完成。不同於其他安全協定，LEAP+首先提出使用多把金鑰來對應無線感測網路多樣的通訊需求，利用相較於 Self-Healing 與 TGDH 較少的能量消耗來達成足夠的安全性。

三、改善 LEAP+的全域金鑰更新

在本篇論文中，我們提出一個方法，可以直接改善 LEAP+中 global key 更新的效能，以下我們先簡單介紹 LEAP+。

3.1.LEAP+的方式

3.1.1.標記法(Notation)

- N：表網路大小中成員數量。
- u 與 v：舉例中兩個主要的成員編號(ID)。
- {f}：隨機函式(pseudo-random function)
- $\{s\}_k$：表示訊息 s 使用金鑰 k 加密。
- MAC(k,s)：訊息 s 使用金鑰 k 加密後的 MAC。

3.1.2.四種金鑰建立方式

在建立四種金鑰之前，每個感測器必須事先寫入幾個會使用到的參數，individual key K_u^m 、運算 pairwise

key 需要用到的 K_i 還有建立 pairwise key 會使用到的計時器(timer)。

○ Individual key

individual key K_u^m 為每個成員 u 所擁有，因為不需要更新所以事先寫入各個感測器當中，主要作為各個成員直接與基地台通訊使用。基地台並不會保留每個成員的 individual key，而是需要使用的時候透過以下算式來取得 $K_u^m = f_{K_s^m}(u)$，其中 K_s^m 為只有基地台知道的金鑰，當基地台需要和成員 u 溝通時，只要馬上算出來就可以用來通訊。

○ Pairwise key

pairwise key 的建立，分為四個階段，分別是金鑰預先分發(Key Pre-distribution)、鄰居的發現(Neighbor Discovery)、金鑰建立(Pairwise Key Establishment)與金鑰消除(key Erasure)。我們在底下較為詳細的描述這四個步驟。

首先在第一階段”金鑰預先分發”(key pre-distribution)中，成員 u 需要透過 $K_u = f_{K_I}(u)$ 來得到自己的 master key K_u，其他網路中的成員亦同，其中 K_I 為事先寫入的運算參數。完成這一步驟之後，每個成員便啟動自己的計時器，並進入第二階段”鄰居的發現”(Neighbor Discovery)。

在第二階段中，成員 u 透過以下動作來發現周圍的成員：

u→*：u, $Nonce_u$

v→u：v,MAC(K_V , $Nonce_u \| v$)

如此便可在不交換任何資訊的情況下完成一次成員 u 與 v 的溝通。網路中每個成員都需要進行這個動作來發現周圍的鄰居。並且在每兩個成員溝通之後建立 pairwise key，也就是第三階段。

”金鑰建立”(Pairwise Key Establishment)，在成員 u 收到 v 回傳的訊息，並且確認這是再收到自己的訊息之後才進行的動作後，便可以透過 $K_{uv} = f_{K_v}(u)$ 來得到 pairwise key K_{uv}。同樣的 v 也可以透過 u 的 ID 來算出 K_u 進而得到 K_{vu}。

直到計時器的時間結束，進入最後一個階段”金鑰消除”(key Erasure)。每個成員消除自己預先寫入 K_I，在這個步驟之後，無法再進行 pairwise key 的建立，且每兩個成員之間都有一把可以跟對方通訊的 pairwise key。

○ Cluster key

pairwise key 建立完成之後，接著是 cluster key 的建立。當成員 u 需要跟周圍的直接鄰居(one-hop neighbor) v_1、v_2、… v_m 建立 cluster key 時，u 需要隨機產生一鉕金鑰 K_U^C，然後用每個鄰居 v_i 的 pairwise key 分別加密之後傳送給各個鄰居，運作如下：

u→ v_i ：(K_U^C) K_{uv_i}

這個建立動作同時也是更新動作，當有新成員加入或是有非法成員被移除之後都需要進行一次，也就是說 cluster key 除了隨著時間規律更新之外，是常常需要作額外的更新動作的。

○ Global key

最後是 global key 的建立，在 cluster key 的建立完成之後，基地台就靠 cluster key 來更新 global key。

BS→ u_i ：M

$u_i \rightarrow v_i$ ：M

M： $u, f_{k_g'}(0), MAC(k_i^T, u \| f_{k_g'}(0))$

M 是基地台廣播出來的訊息，內容包括要被移除出網路的非法成員 u，新的 global key k_g' 還有 verification key $f_{k_g'}(0)$ 與下個時脈會公佈的 MAC key k_i^T，在這裡我們不詳細說明其它金鑰的用途，我們只要知道這個訊息 M 含有非法成員的 ID 與新的 global key 有關的更新資訊。

global key 除了一開始預先寫入作為初期使用的第一把 global key，之後

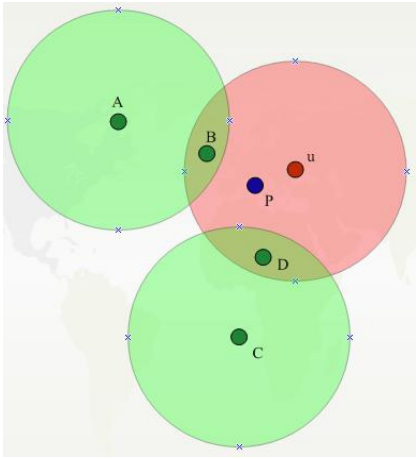
所有的 global key 都需要更新，尤其是在有非法成員離開或是新成員加入的時候。在知道有非法成員的存在後，基地台會廣播一個附有非法成員名稱的訊息，如果自己的鄰居當中有此成員的存在，便刪除與此成員溝通的 pairwise key 和 cluster key。在更新的時候便可以防止非法成員收到任何新的 global key 的相關資訊或是理解新的 global key。

3.1.3.LEAP+法的問題

○ LEAP+之假設

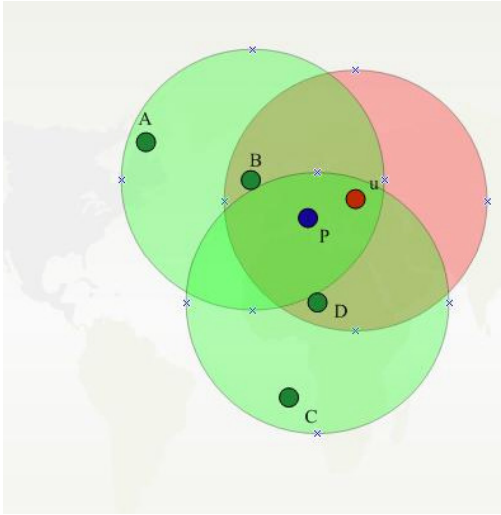
1. LEAP+假設當網路中有任何成員被綁架可以馬上偵測到並且通知基地台，基地台也會馬上開始進行應對的措施。
2. LEAP+假設基地台不會被攻擊成功，並且擁有無限的能量與運算能力。
3. 假設感測器已經按照 LEAP+的安全協定完整的建構成一個可以正常運作的無線感測網路。
4. LEAP+假設敵人可以從任何開放的管道進行任何種類的攻擊。

LEAP+的 global key 更新方式可以用簡單的用圖形來表示，圖一中 A、B、C、D 和 P 點分別代表正常的成員，u 點代表將被移除的非法成員。在基地台發現了 u 的存在之後，會啟動排除非法成員的程序－廣播訊息 M，我們假設這個區塊 M 訊息是由 A 與 C 先收到：



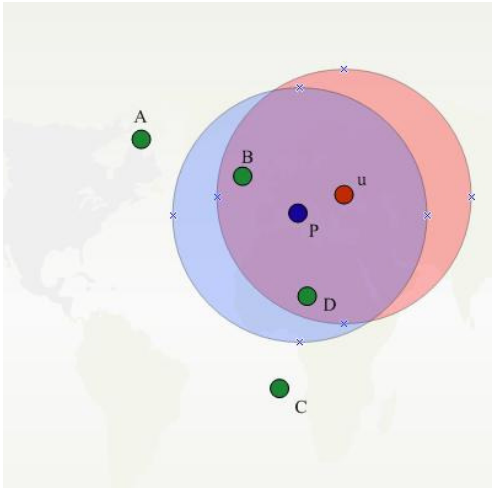
圖一、LEAP+(1)

在 A 與 C 收到訊息 M 之後首先利用來源端的 cluster key 解密 M，再用自己本身的 cluster key 加密後廣播。A 的廣播範圍內有 B 點，C 的廣播範圍內有 D 點，接著 B 與 D 進行下一次的廣播動作。



圖二、LEAP+(2)

B 收到 M 之後，利用 A 的 cluster key 解密並且用本身的 cluster key 加密繼續做廣播。同樣的 D 收到 M 後用 C 的 cluster key 解密後用自己本身的 cluster key 加密繼續廣播。同時 B 與 D 再送出訊息 M 之後，馬上刪除與 u 的 pairwise key 並且更新自己的 cluster key，如圖二。接著 P 與 u 會收到訊息 M。



圖三、LEAP+(3)

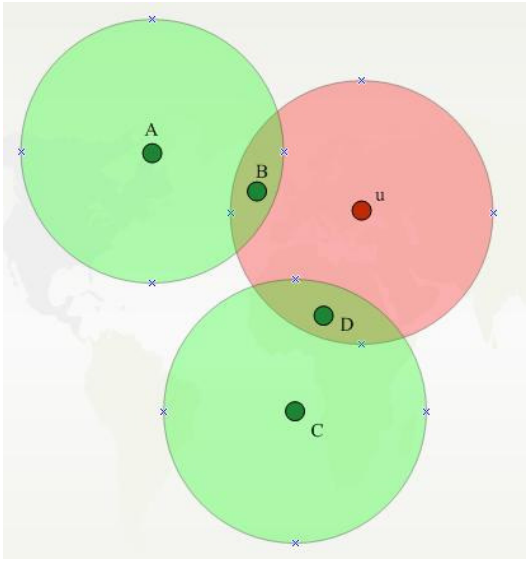
P 收到之後一樣按照前面的步驟進行 M 的向前傳送，傳送後刪除與 u 的 pairwise key 並且更新自己的 cluster key，如圖三。

我們可以發現原本的 LEAP 有一個很大的問題存在：只要與 u 有直接溝通能力的成員就算可以馬上切斷與 u 的關係，但仍必須付出相當的代價來確保安全－刪除 pairwise key 後更新 cluster key，其中更新 cluster key 這個舉動會造成相當高的能量消耗，從這個事件來看，需要刪除 pairwise key 的成員有 B、D 與 P。由於更新 cluster key 所需的運算能量與該成員的分支數成正比，即使在一般的成員密度下，這也是相當浪費能量的方式，就算敵人不做任何竊聽攻擊，只要透過不斷的綁架合法成員並且故意被發現，頻繁的更新 global key 就足以讓整個網路的壽命(life time)縮短。因此我們提出以下方法，透過簡單的技巧便可以在不啟動 cluster key 更新機制的的情況下安全的更新 global key。

3.2.改善 global key 更新的效能

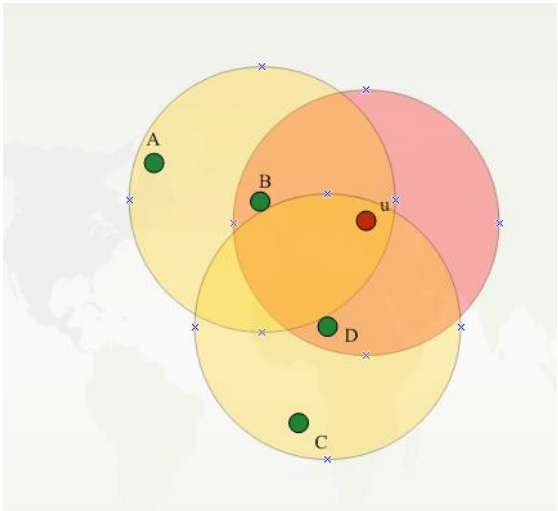
在 global key 的更新上，我們首先利用到 LEAP+本身建立的記憶體內資訊，也就是所有的直接鄰居(one-hop neighbor)名單，且我們利用一個額外的訊息來防止非法成員 u 繼續接收到新的 global key 相關資訊，我們將這個訊息稱作 S：STOP_message，其中含有非法成員 u 的 ID，透過這個訊息的使用，與 u 有 pairwise key 直接相連的合法成員不需要因為 M 的到來而被迫更新 cluster key。簡單的說，我們不需要作額外的運算，而是利用感測器內記憶體的資訊來完成這項工作，下面我們來看這個方法詳細的流程。

3.2.1.效能的改善-正常情況



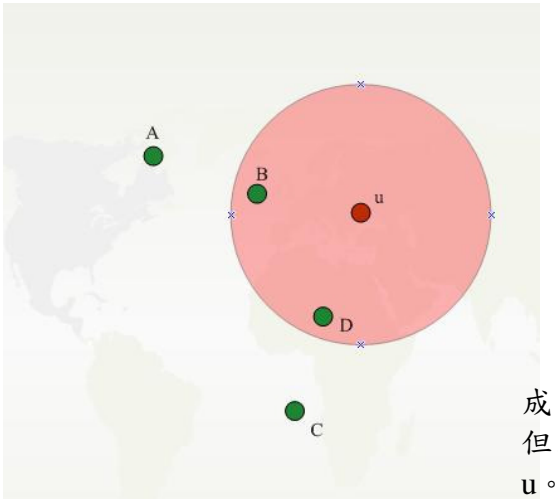
圖四、一般情況(1)

如圖四表示，從 A 與 C 收到訊息 M 開始，使用來源端的 cluster key 解密並且用自己的 cluster key 加密後繼續做廣播。B 與 D 將是下一階段會收到 M 的合法成員，此時我們需注意到 B 與 D 是非法成員 u 的直接鄰居。



圖五、一般情況(2)

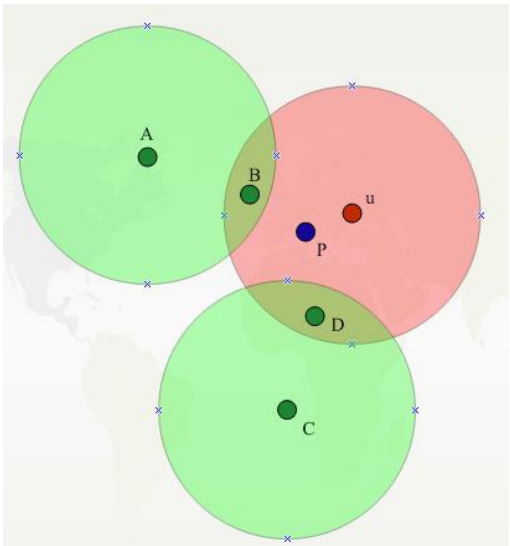
B 與 D 收到 M 之後，得知 u 是這次更新要排除的非法成員，並且在自己的直接鄰居名單中，使用 S 訊息取代 M 訊息，也就是說 u 收到的是一個 STOP 訊息，其中並不包含任何新 global key 的訊息。這在這更新 global key 的步驟之後 B 與 D 也不需要更新 cluster key，只需要刪除與 u 的 pairwise key 就可以了，如圖五。



圖六、一般情況(3)

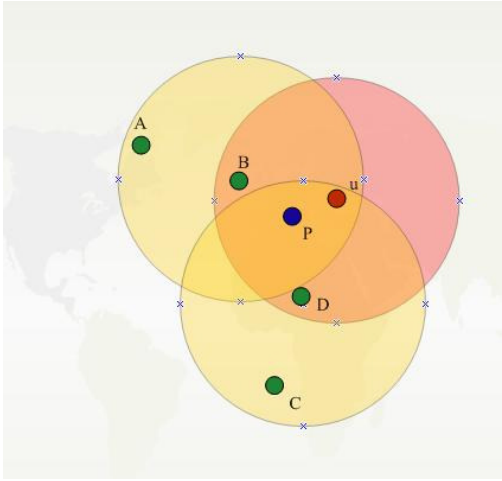
更新完成之後使用新的 global key 發布訊息，在遇到非法成員 u 的時候可以用以上步驟來防止 u 收到任何有關的資訊，如圖六。

3.2.2.效能的改善-特殊情況



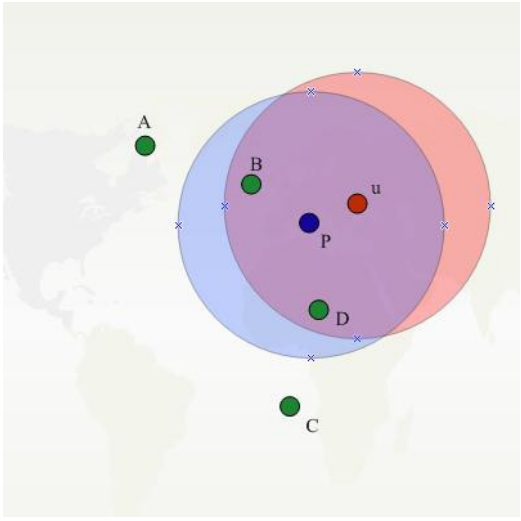
圖七、特殊情況(1)

在我們的方法下會有一種特殊情況發生，我們在這裡特別說明，也就是圖七中的合法成員 P 存在的情況。同樣的 A 與 C 收到 M 之後繼續廣播傳送。



圖八、特殊情況(2)

B 與 C 收到 M 之後知道 u 是非法成員，因此不廣播 M。改廣播訊息 S，但因為 P 點的直接鄰居只有 B、D 和 u。B 與 D 都只發送 STOP 訊息。合成員 P 發現自己的三個直接鄰居 B 與 D 接傳來 S 訊息，並且從 S 訊息中得知 u 是將被排除非法成員，如圖八。



圖九、特殊情況(3)

合法成員 P 可以利用 pairwise key 向合法的直接鄰居來索取訊息 M，如此完成一次特殊情形下的更新，如圖九。

四、分析

4.1.安全性

在無線感測器網路中，敵人可以進行的攻擊包括欺騙(spoof)攻擊、竄改(alter)攻擊、重送(re-play)攻擊、選擇傳送(selective forwarding)攻擊、氾濫訊息(flood)攻擊、複製偽裝(sybil)攻擊、排水孔(sink-hole)攻擊法與蟲洞(worm-hole)攻擊法等多樣的攻擊。一個正常運作的 LEAP+可以利用四種金鑰來阻擋以上的所有攻擊。尤於我們對於 LEAP+本身的運算與架構並沒有作大量的修改，因此我們的方法並不會影響到 LEAP+原有的安全性，因此在安全性的部份與 LEAP+原有的安全性相同，同樣可以抵擋目前已知的各種攻擊。

4.2.效能評估

在效能評估的部分，我們先分析傳統 LEAP+的三個成本，分別是運算成本，通訊成本還有空間需求，然後在分析後說明我們的方法如何有效的節省成本。

4.2.1.運算成本

在 LEAP+當中，有成員被綁架了，所有合法成員需要先用 pairwise key 加密來更新 cluster key。 所以鄰居的數量決定加密的次數，也可以說是網路的密度決定加密的次數。我們假設 n_0 被綁架，, n_i (i=0,1,2,3,…)是所有的鄰居，則加密的成本 $S_e = \sum_{i=1}^{n_0} n_i$ ，解密亦同，在最糟糕的情況下約需要加密 $(\mathrm{Max}(n_i)+n_0-1)$ 次。要是網路的大小為 N，則平均的情況下更新 cluster key 的成本為 $\frac{2S_e}{N}$。從這裡可以看出網路的大小會直接影響 LEAP+的 global key 運作成本，又每個成員都需要加解密各一次，故成本為 2N。

總和來看，一個網路大小 N，分支度為 d 的情況下，每一個成員平均需要成本為 $\frac{2(d-1)^2}{(N-1)+2}$ 。

LEAP+在每一次更新 global key 之前需要先進行 cluster key 的更新，在我們的方法中 cluster key 的更新機制並不會因為 global key 更新而啟動，而是在系統設定的時間下規律的更新，相較於 LEAP+因非法成員出現就更新 cluster key 的成本，我們的方法可以確保 global key 的更新不會影響 cluster key 的更新，簡單的說就是將更新 cluster key 的運算成本獨立出來，不會受到 global key 更新的影響。若是非法成員 u 有 d 個直接鄰居，則每一次 global key 的更新至少可以節省一次 cluster key 的運算成本，也就是每個直接鄰居可以節省 d 次 pairwise key 的加密。

4.2.2.通訊成本

對 global key 來說，通訊成本和運算成本相似，皆為 $O(\log N)$。對 cluster key 來說通訊成本為 $\frac{(d-1)^2}{(N-1)}$ 。在有 1000 的成員的網路中，若分之度 d 為 20，則要完成一次非法成員撤除動作，也就是一次傳送與一次接收，通訊成本為 2.8 個 key。

我們的方法可以有效的把 global key 的更新機制與 cluster key 的更新機制分開，變成兩個獨立的更新動作。而不是 LEAP+原本的 global key 更新啟動 cluster key 更新機制，可以有效的節省通訊成本，可以直接省去每次附加在更新 global key 之後的 cluster key 更新成本 $\frac{(d-1)^2}{(N-1)}$ 。

4.2.3.空間需求

我們假設某合法成員有 d 個合法鄰居，則它需要保存 1 把 individual

key、d 把 pairwise key、d 把 cluster key 與 1 把 global key。在我們的方法中，存放的鑰匙數和 LEAP+是相同的。

成本 / 方法	更新 global key 的平均 **運算成本**	更新 global key 的平均 **通訊成本**
LEAP+	$\frac{2(d-1)^2}{(N-1)+2}$	$O(\log N) + \frac{(d-1)^2}{(N-1)}$
我們的方法	$\frac{2(d-1)^2}{(N-1)+2} - \frac{2S_e}{N}$	$O(\log N)$

註解:
1. d：感測器的分枝度
2. N：網路大小
3. S_e：加密的成本

五、結論

在本篇論文中，我們提出一個利用感測器記憶體內既有的資訊來改善 LEAP+能量消耗的方法，主要貢獻在於 LEAP+中 cluster key 的更新不會因為 global key 的更新而被強迫進行更新，cluster key 可以按照原本網路系統的設定時間來規律的進行更新。如此一來可以省去多餘的 cluster key 更新運算成本與通訊成本，並且不會影響到 LEAP+原有的安全性與空間需求。

此法更可以套用到其它使用 global key 的無線感測器網路安全協定中，並且同樣能夠節省相當的能量。

六、參考文獻

[1].Zhu, S., Setia, S., & Jajodia, S. (2006). LEAP+: Efficient security mechanisms for large-scale distributed sensor networks. *TOSN*, 2(4), 500-528.

[2].Zhu, S., Setia, S., & Jajodia, S. (2003). *LEAP: efficient security mechanisms for large-scale distributed sensor networks*. Paper presented at the ACM Conference on Computer and Communications Security.

[3].Perrig, A., Szewczyk, R., Tygar, J. D., Wen, V., & Culler, D. E. (2002). SPINS: Security Protocols for Sensor Networks. *Wireless Networks*, 8(5), 521-534.

[4].Karlof, C., Sastry, N., & Wagner, D. (2004). *TinySec: a link layer security architecture for wireless sensor networks*. Paper presented at the SenSys.

[5].Heo, J., & Hong, C. S. (2006). Efficient and Authenticated Key Agreement Mechanism in Low-Rate WPAN Environment. *Wireless Pervasive Computing, 2006 1st International Symposium on*, 1-5.

[6].Staddon, J., Miner, S., Franklin, M., Balfanz, D., Malkin, M., & Dean, D. (2002). Self-healing key distribution with revocation. *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on*, 241-257.

[7]. Steiner, M., Tsudik, G., & Waidner,

M. (1996). *Diffie-Hellman key distribution extended to group communication*. Paper presented at the CCS '96: Proceedings of the 3rd ACM conference on Computer and communications security, New York, NY, USA.

[8].Li, L., Li, J., Wu, Y., & Yi, P. (2008). A Group Key Management Scheme with Revocation and Loss-tolerance Capability for Wireless Sensor Networks. *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on*, 324-329.

[9].Gerelbayar, T., & Park, J. S. (2007). A New Centralized Group Key Distribution and Revocation in Sensor Network. *Computational Intelligence and Security, 2007 International Conference on*, 721-724.

[10]. Hong, T., Liehuang, Z., Yuanda, C., & Dazhen, W. (2008). A Novel Tree-based Authenticated Dynamic Group Key Agreement Protocol for Wireless Sensor Network. *Electronic Commerce and Security, 2008 International Symposium on*, 540-544.

[11]. Poornima, A. S., & Amberker, B. B. (2008). A Secure Group Key Management Scheme for Sensor Networks. *Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference on*, 744-748.