Virus Impossible

CIH<Software Magician>

EMAIL：**IamCIH**（AT）**gmail**（DOT）**com**

GIGA-BYTE TECHNOLOGY CO., LTD.

2009/07/18

HIT Conference 2009

這一天

『怎會這樣？我的電腦被駭客入侵了！什麼？螢幕出現倒數10秒，就要把我電腦整個資料破壞！』

螢幕出現倒數：10~9~...1~0秒
最後出現一串字：*Just Kidding*

『本台記者報導：今天下午，在同一瞬間，全世界電腦都出現這個訊息，"*You've been hacked*"。幸好，駭客最後沒有真正破壞電腦資料。但是已經引起全世界極度惶恐，從未有這樣事情發生過。各國政府機關、資安單位全面關切中。駭客的下一步是什麼？』

HIT Conference 2009

前言

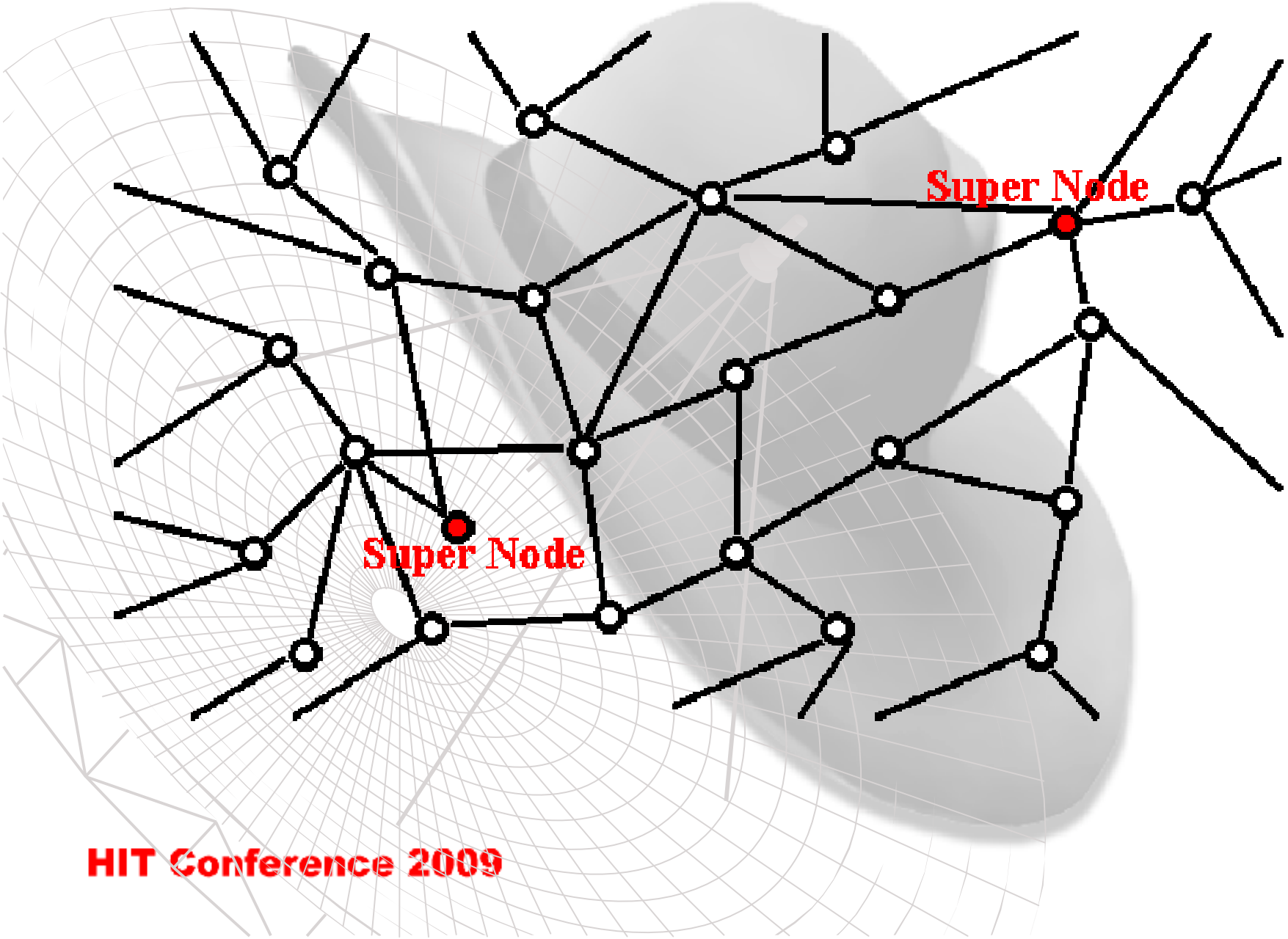
這篇文章的想法，只是一個很初步的簡單雛形，並沒有實際驗證過。也或許藏著一些不可能實現的技術，或是錯誤的想法。就以技術討論的話，一定會有更大的改善空間。裡面的論點，都基於理想狀態，沒有細節分析所有可能性。若是要完整實現整個系統，將是非常複雜的事情。因此，此篇內容，只是提供大家，另一種病毒可能行為的思考。

行為模式

- 病毒
- 蠕蟲
- 感染行為
  - 鑽網路漏洞
  - 感染執行檔
- 殭屍網路(BotNet)
- P2P
- 動態執行Function

基本背景

- **Super Node**：中毒電腦，具有**Public IP**。
  - 建立**Node**彼此間的連線(協助穿越**NAT**)。
  - 除此之外，就跟一般**Node**完全相同。
- **Node**：所有全部的中毒電腦(包括**Super Node**)。
- 一般**Node**：不包括**Super Node**的**Node**。
- 這些**Node**，彼此會有**2~5**個連線，交集成一個病毒網路系統。



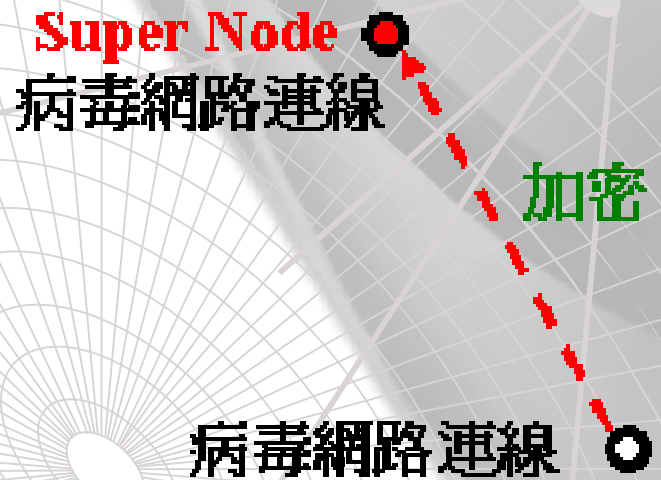
病毒的架構

- **Function Layer**
  - 接收Command，若沒有這個Command所需要的Function，就會去病毒網路系統尋找。
- **Network Packet Layer**
  - 接收來自Function Layer，所有病毒封包的處理、加密、傳遞工作。

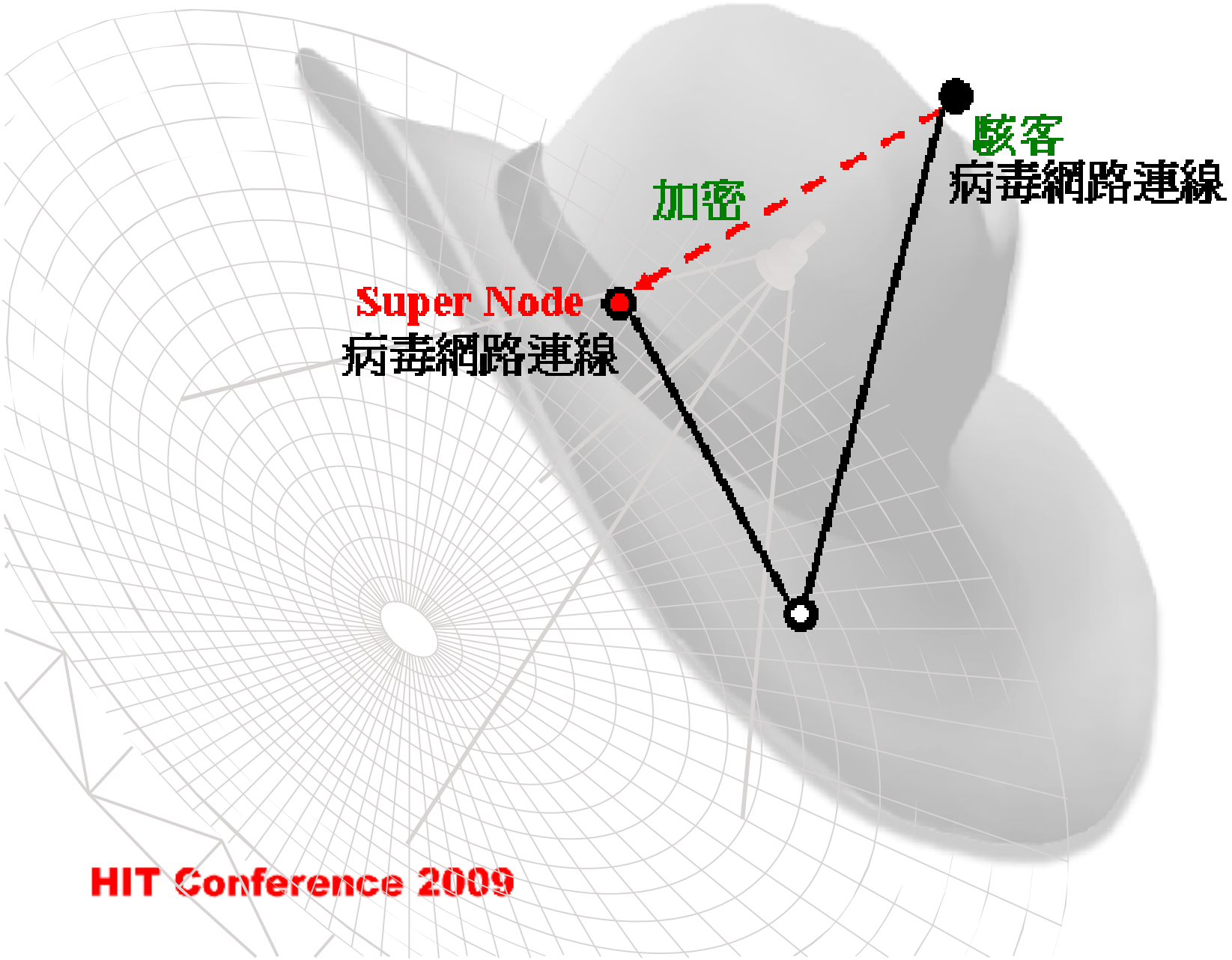
此病毒，一開始是完全沒有任何感染能力，也就是沒有"感染"的Function。

- 最原始的病毒，只有1個Function：病毒網路連線

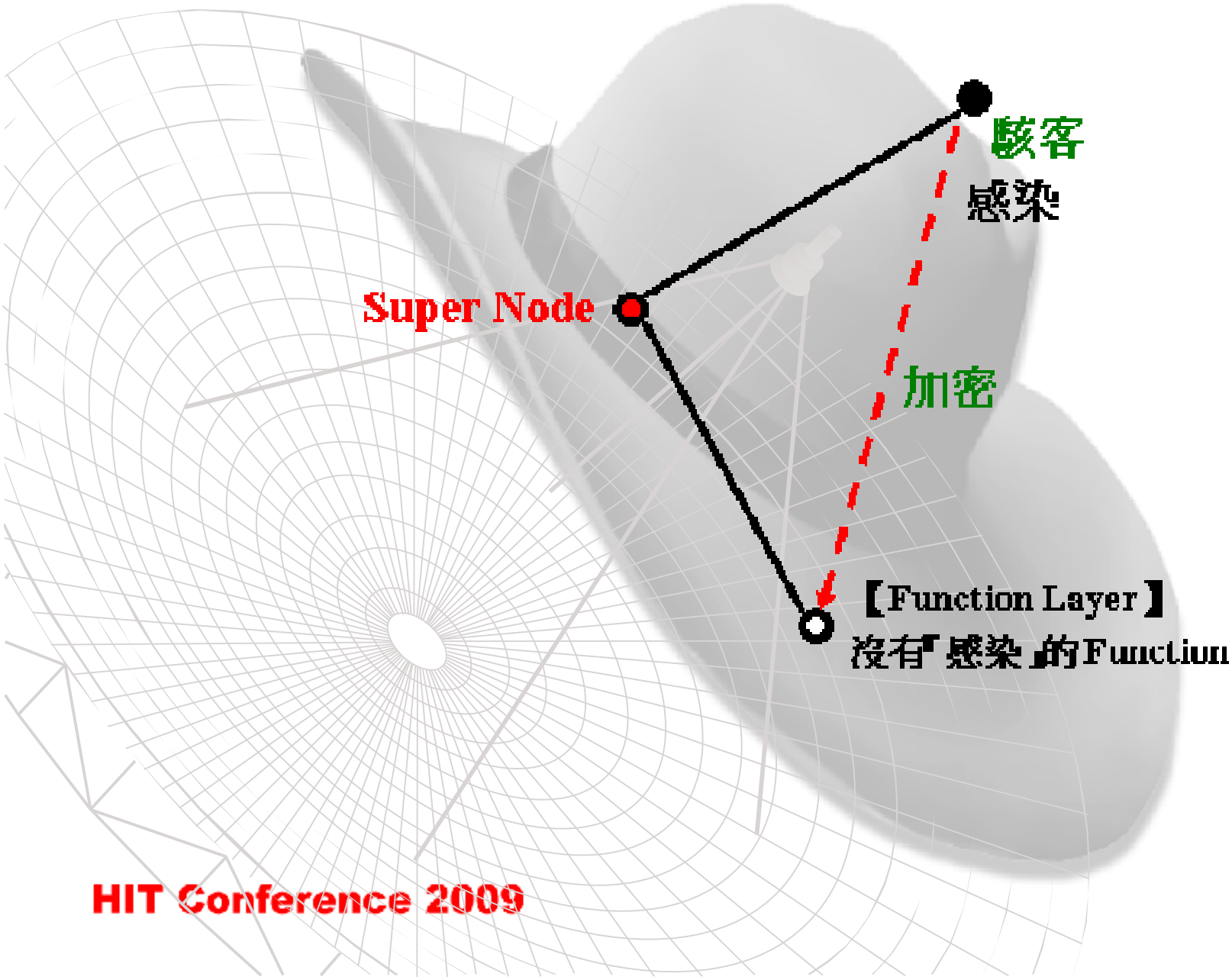
病毒運作的過程



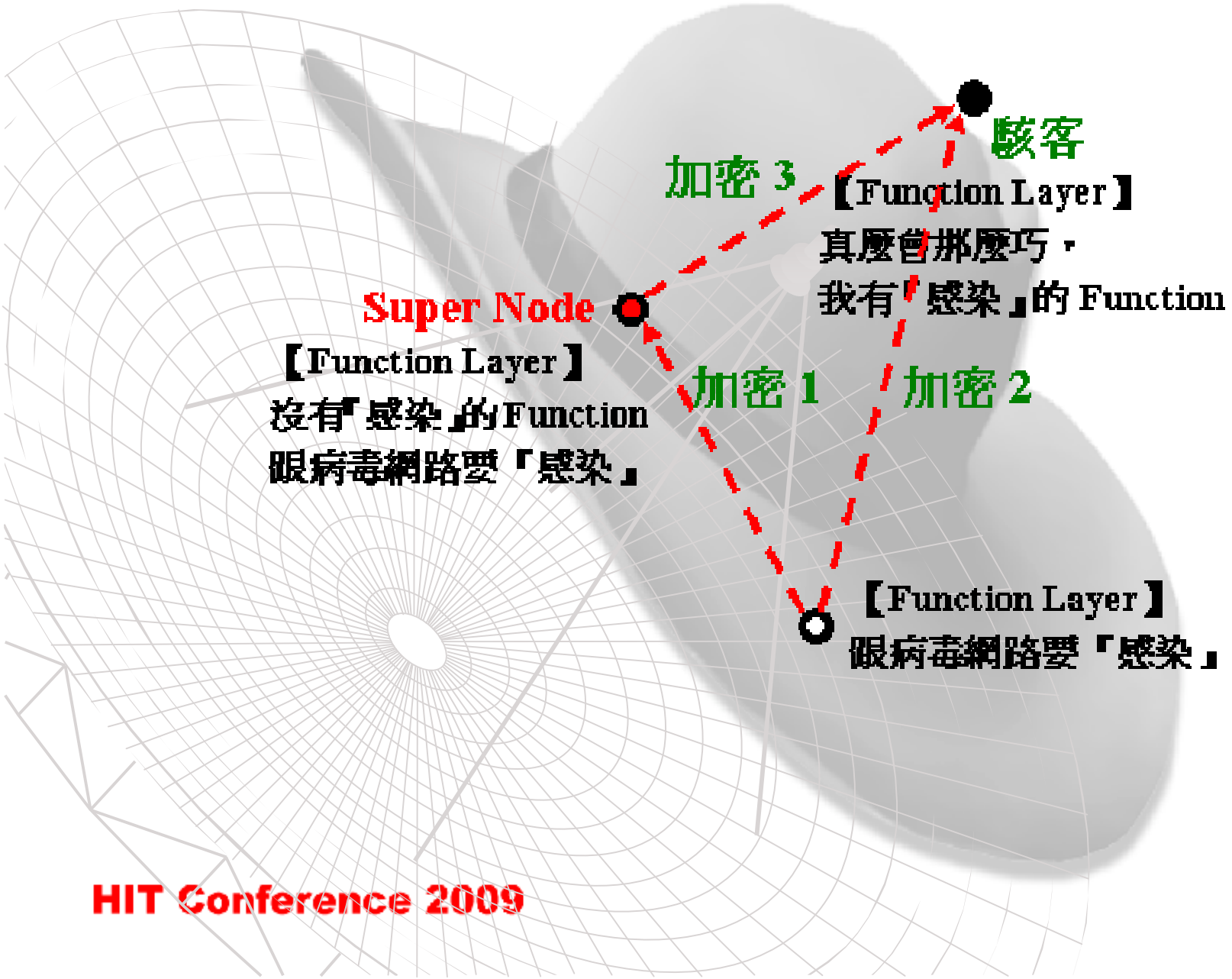
HIT Conference 2009



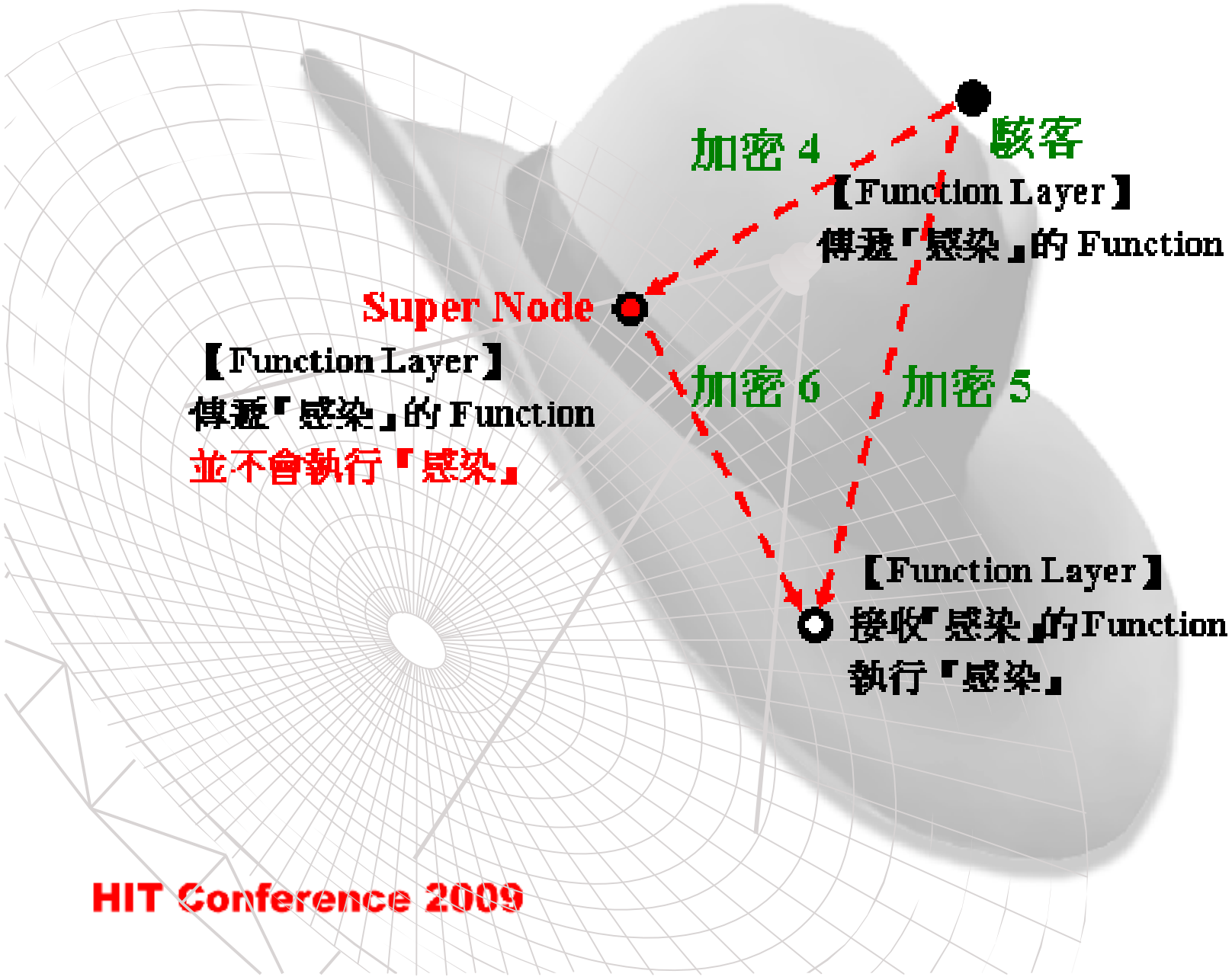
HIT Conference 2009



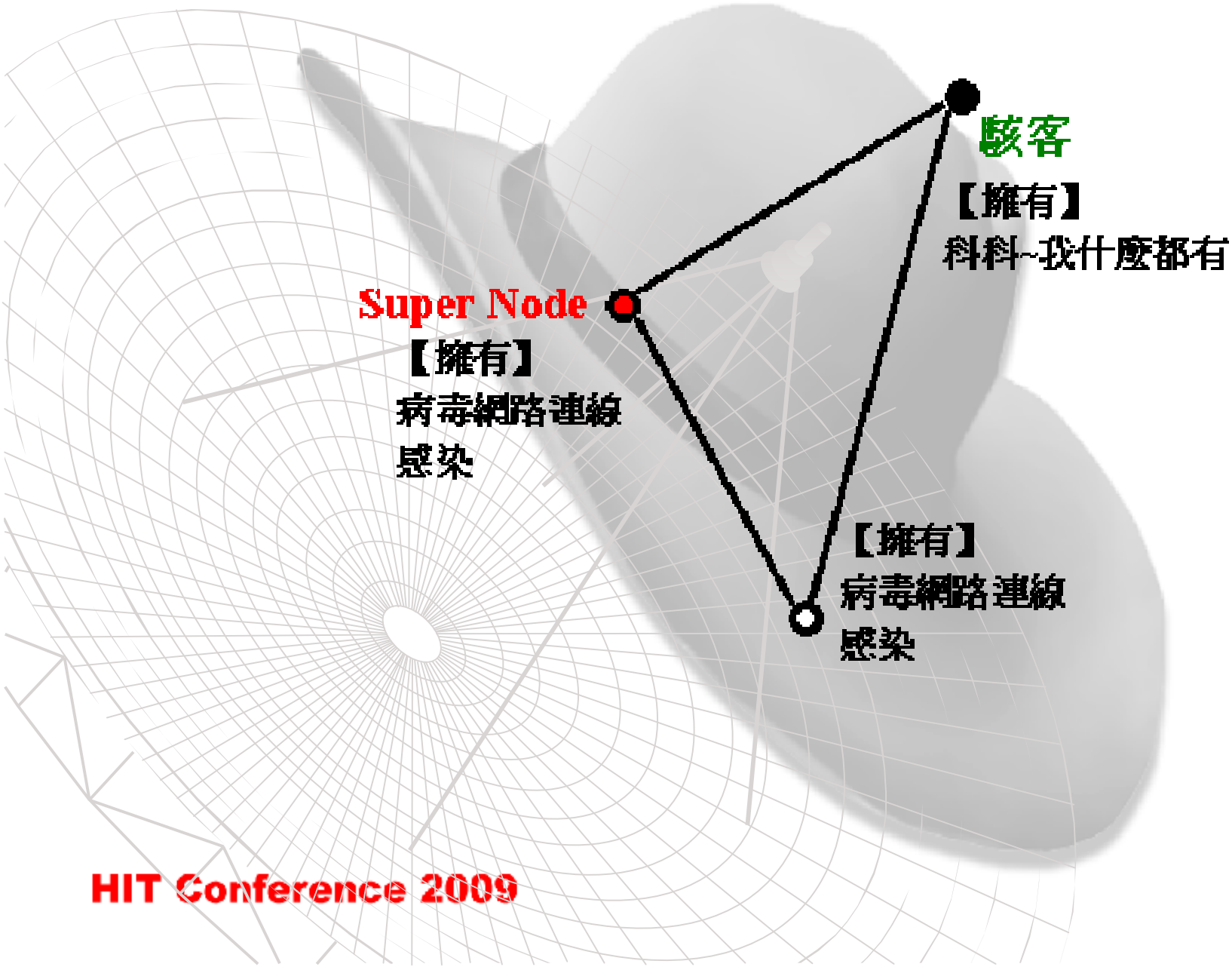
HIT Conference 2009



HIT Conference 2009



HIT Conference 2009



HIT Conference 2009

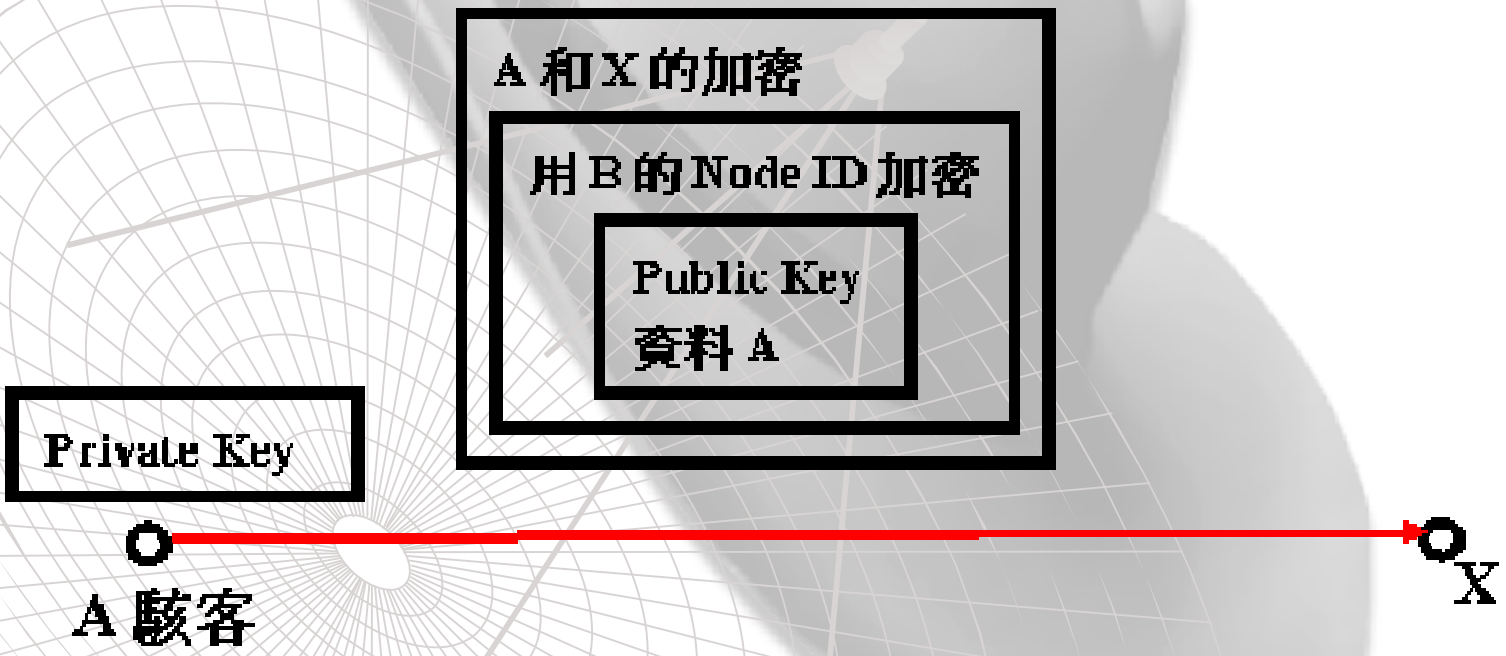
中毒檔案的內容

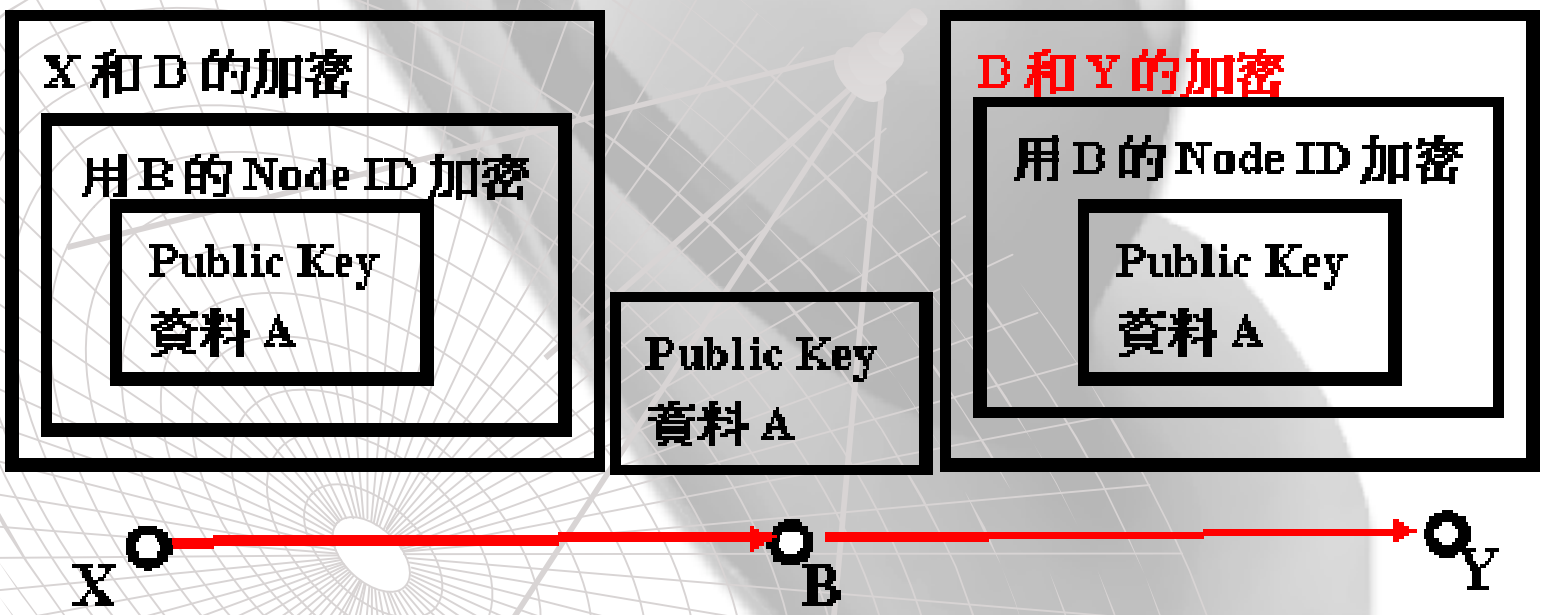
- Function Layer程式碼
  - Function：病毒網路連線
- Network Packet Layer程式碼
- 所有Super Node的IP(一開始是預設1~2個，之後隨著病毒的擴散，系統會自動增加)。

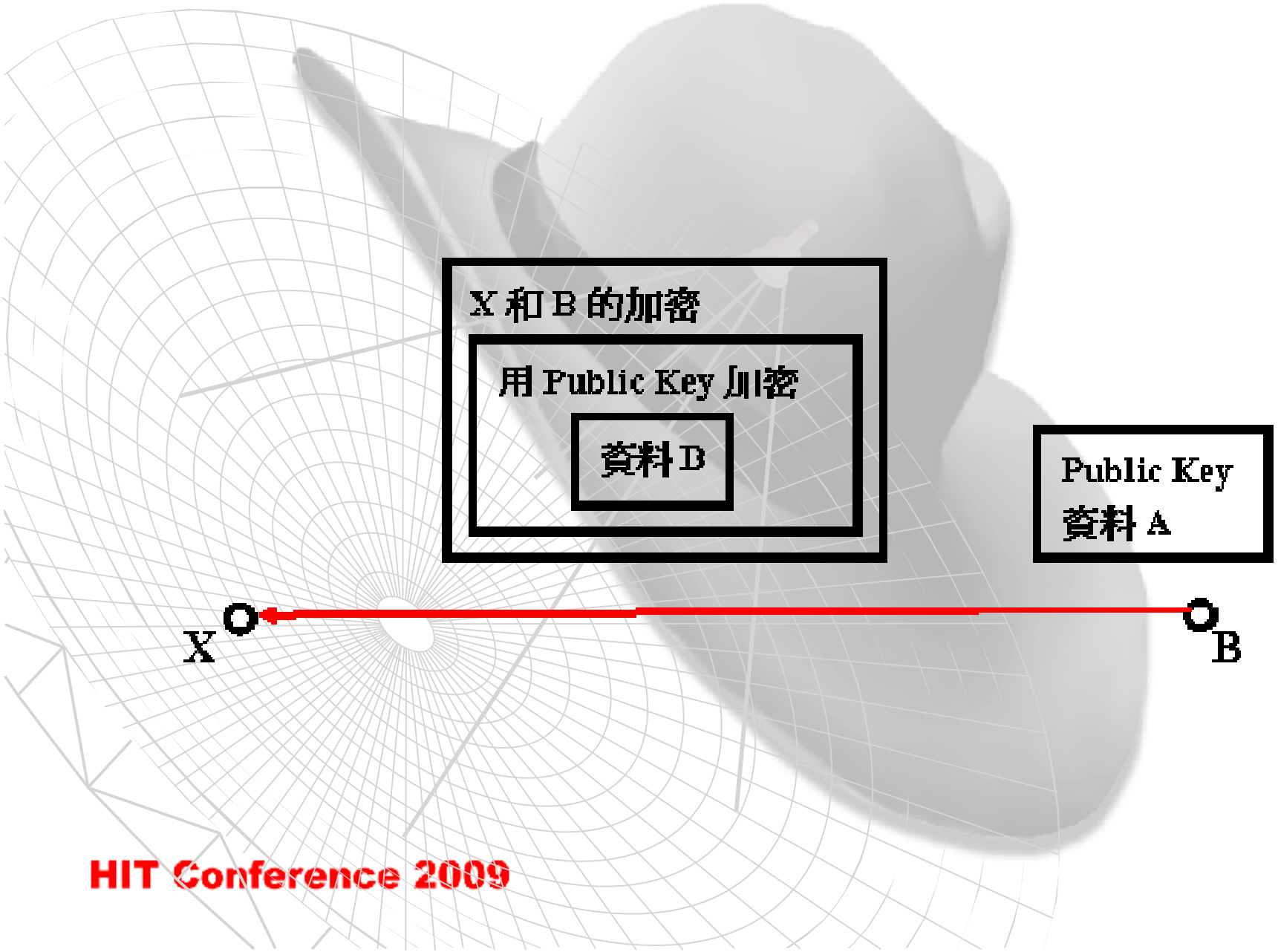
基本技術探討

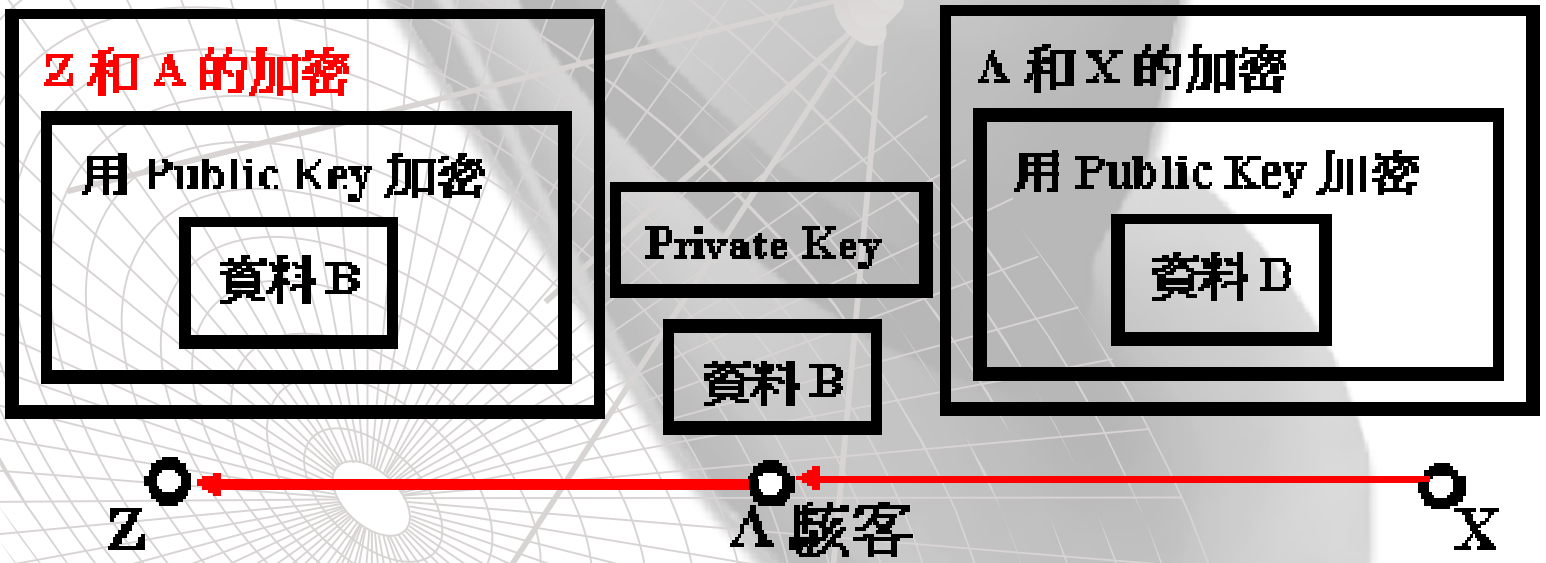
- 病毒程式碼
  - 在kernel mode下執行。
  - 全部用組合語言。
  - 所需要的動態Function，也是用組語，可以直接放在記憶體上執行。
- 每個Node會有唯一的"Node ID"(例如：可以用MAC、或是其他方法產生)。
- 封包會有生命週期，以避免無限在網路上面跑。

A駭客，送封包給B。B收到後，回封包給A駭客：





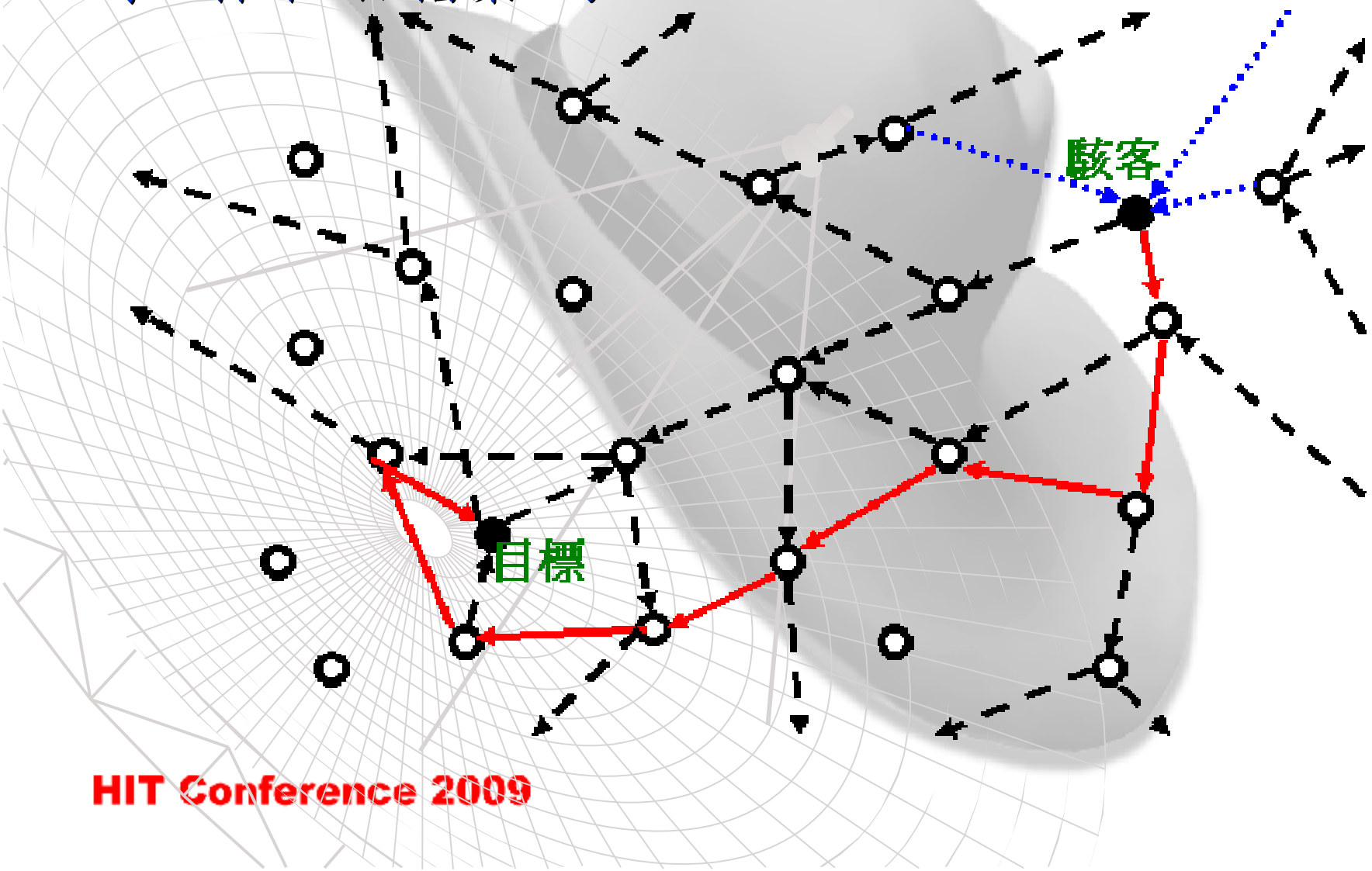




HIT Conference 2009

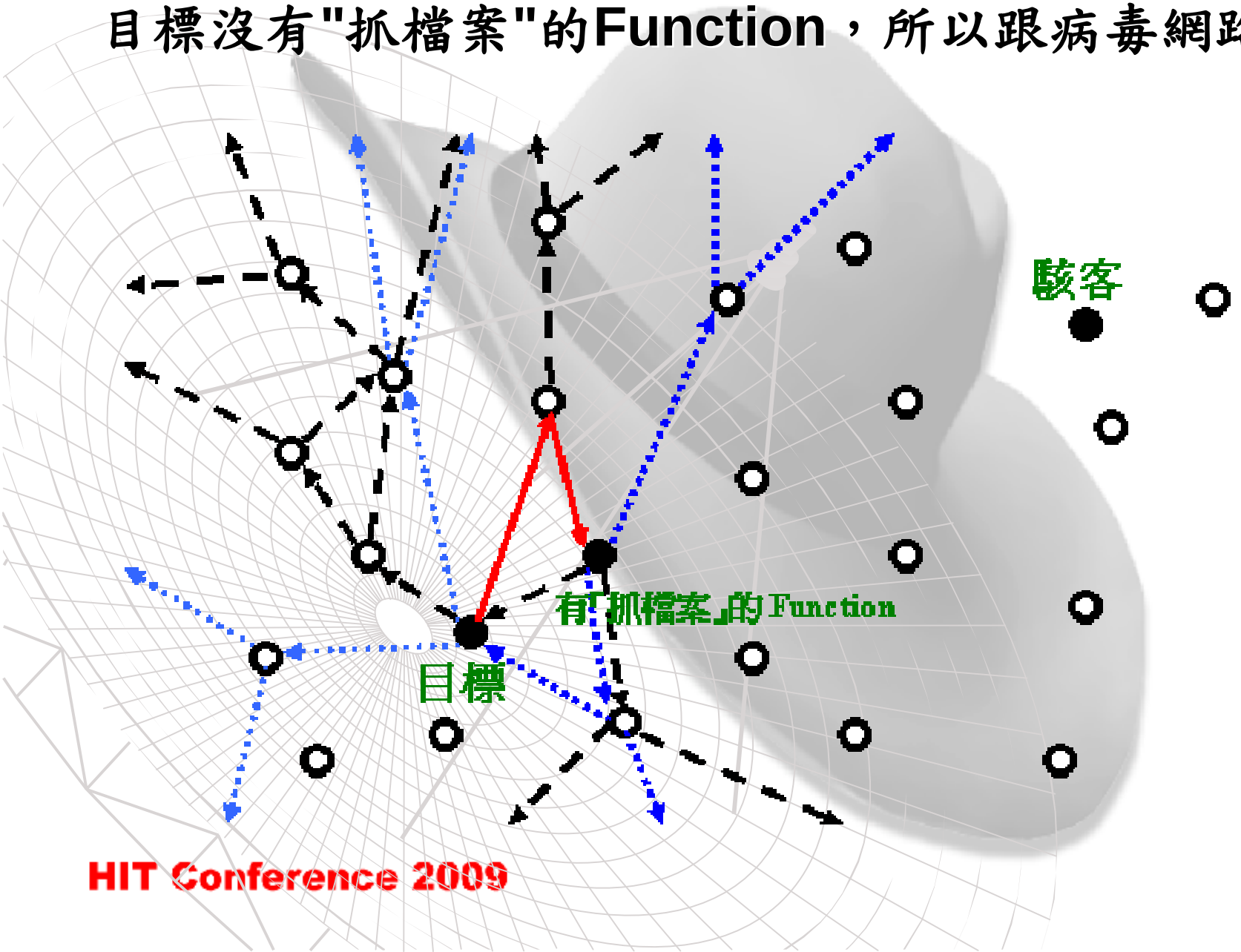
Hacker遠端控制

對目標下"抓檔案"的Command

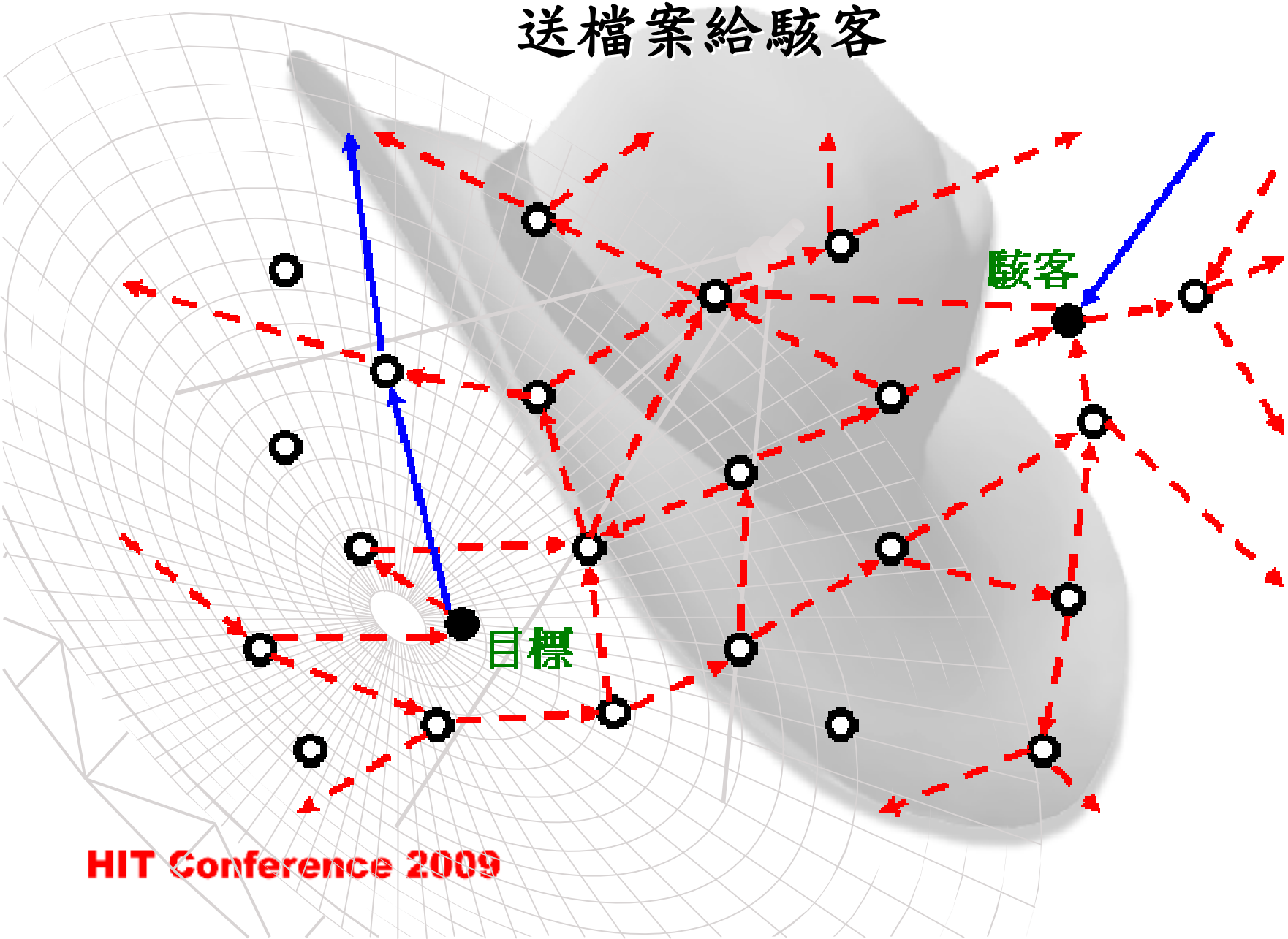


HIT Conference 2009

目標沒有"抓檔案"的Function，所以跟病毒網路要

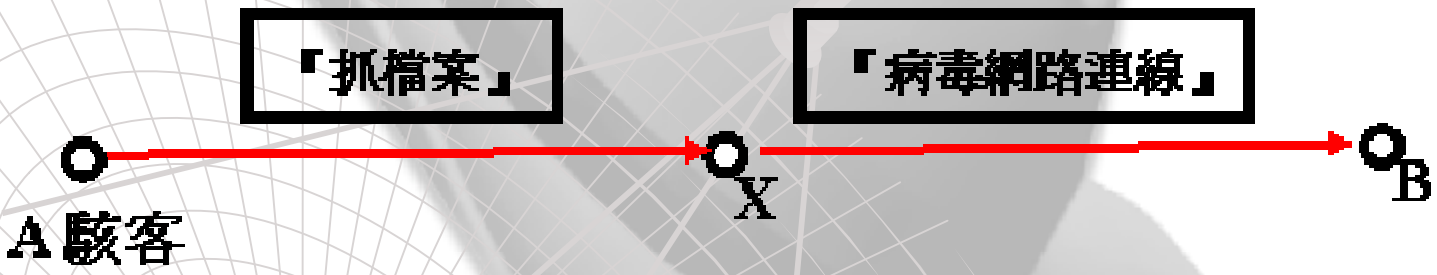


送檔案給駭客



防止Hacker被抓

- 病毒網路系統，採取Broadcast封包傳送。
  - 傳送者，可能會收到自己的封包。避免變成封包傳遞的起點。
  - 接收者，藏在一盤沙裡面的一粒沙。
- 封包加密。
- 封包傳遞過程，遇到1個假的Node。
  - 駭客的目標，或許有可能被找出來。(當所有Node ID被得知時)
  - 駭客本身，被找出來的可能性，是微乎其微。(因為需要那一把駭客產生的Private Key)
- 封包的傳遞，採取非即時性，隨機緩慢性，以避免被追蹤。
  - 每個Node送封包，亂數delay，1秒~10分鐘。
  - 避免封包，同一時間大量傳輸。
  - 整個封包傳遞過程，所有不同類型的封包，亂七八糟混在網路。幾乎很難，從時間關係追出封包流向。



總結：

Hacker很有耐心。就算他要抓某一台電腦的一個檔案，等一個禮拜，他都可以等。

需要思考的問題

- 所有Node如何完整串接在一起，避免變成兩個以上的病毒網路？
- 當其中一個Node消失後，其它Node如何串聯在一起？
- Broadcast封包傳送的演算法，是否會導致某些Node永遠收不到封包？(因為封包有生命週期)
- 什麼樣的網路拓撲(Topologies)最適合這個系統？
- 還有其他更多...