Notes of Cryptography

Squirrel

March 2, 2025

Preface

Course

密碼學設計與分析 Cryptography Design and Analysis (11320IIS500900) in NTHU

1 L1

1.1 Merkle 的故事

Merkle 在大學部修了一個課，然後要交一個 project。他在交這個作業的時候，提到了 Public Key Cryptography 的想法。當時的導師並不看好這個東西，所以 reject 了，最後他也退掉了這門課。之後他找到另一個很欣賞他的老師，覺得應該要「Publish it, win fame and fortune」，所以他將這篇文章那個投到了 CACM (Communications of the ACM)。第一次投期刊就因為「這個想法不是當今的主流想法」而被拒絕。在 Merkle 的某些堅持之下，過了快三年終於讓 CACM 接受了這篇文章。

這邊的故事及當時的論文，可以在 https://ralphmerkle.com/1974/ 找到。

另外影片中的 link 有誤，應該改成 https://ralphmerkle.com，不然你只會找到一間搞 CRM 和賣資料的公司。

1.2 Conventions

- 離散且有限的時間 (discrete and finite world)
  ⇒ 因為我們正在討論 computer science
- Data v.s. Informatition
- Machine (function/algorithm) 需要在 polynomial time 下執行
  ⇒ 因為我們需要能在一定時間內看到結果，不想要等到天荒地老
  ⇒ 不一定**強制**要求 polynomial time，但這堂課大部分會是這樣
- Alice and Bob：就是 sender 和 receiver，通常是 Alice 要傳訊息給 Bob
  ⇒ 還有其他角色，可以參見 Wikipedia：
  https://en.wikipedia.org/wiki/Alice_and_Bob
- 計算 (computation)：任何遵循 well-defined model (例如 algorithm、protocol) 的 calculation。
- Efficiency
  Input size: $|x| = n$ bits
  其他的就是拿 complexity 概念來作為 efficiency 的概念
- Crypto 像是信仰 (Faith) ?
  密碼學不一定總是對的，但我們需要相信某些東西才能繼續在密碼學上前進
  這些東西包含：
  ⇒ 某些數學問題很難被解決
  ⇒ 某些假設無法被打破 (通常指在 poly-time 底下)
  ⇒ 某些底層的密碼工具 (underlying crypto primitives) 是安全的
  ⇒ P $\neq$ NP
  ⇒ 亂數/隨機 (randomness)，因為我們不知道真的亂數長什麼樣，所以無法驗證

1.3 Overview

§ 什麼是密碼學？

如果我們不在意安全，那麼我們不需要密碼學。
(If do not care security, we won't need crypto.)

安全 (security) 可以由以下兩點來定義：

- 目的 (purpose)：我們需要達到什麼效果
- 需求 (requirements)：為了達到目的，我們需要達成哪些目標

一些密碼學相關的內容：

- 加密 (entryption)
- 數位簽章 (signature)
- 零知識 (zero knowledge)
- 安全計算 (secure computation)

1.4 Notations

Private key encryption (or "secret key encryption")

就是對稱式加密，加密和解密皆使用同一個 key

Public key encryption

公鑰系統。一個公鑰會對應一個私鑰。公鑰會公開，私鑰不公開。
若 Alice 要傳訊息給 Bob，則 Alice 會使用自己的公鑰加密，並且讓 Bob 使用「與 Alice 的公鑰相對應的」私鑰進行解密。

Zero knowledge

A 想向 B 證明某件事情，但不想透漏任何其他的額外資訊。
Ex1：我想向你證明我有 100 萬，但不想真的放 100 萬現金在你眼前 (以免被你搶走)，所以我可以要求銀行開立證明來達到這個目的。Ex2: 我想向你證明我真的知道「威利在哪裡」。我可以用一張比原圖更大張的紙，並且在上面挖一個威利形狀的洞，以此來達到目的。

1.5 Story of solving impossibility

(這邊的例子經過一點點調整)
你的上司要求你解決一個問題 Q，並且告知你如果無法解決問題就會被炒魷魚，並被另一個比你聰明的傢伙取代。你雖然不知道怎麼解決 Q，但你知道另一個**相關的**知名問題 $\widetilde{Q}$ (Q tilde) 在現今根本就沒人會解。最後你告訴你的上司，由於「現在根本沒人知道如何解 $\widetilde{Q}$」，所以「也沒人會解 Q」，因此這問題解不了，而另一個自稱聰明的傢伙其實是騙子。

重點就是

If there's a good algorithm for Q, then there exists a good one for another well-known problem $\widetilde{Q}$.
這句話的逆否命題就是
If there's no algorithm for $\widetilde{Q}$, then there's no algorithm for Q either.

這背後的概念就是 reduction (就演算法的那個 reduction)。

1.6 Principle of modern crypto

Kerckhoff's principle

「加密方法不能被要求是保密的，就算它落入敵人手中也不應該造成麻煩」
意即，整套加密方法的安全性只仰賴金鑰的保密。

(原文：It should not require secrecy, and it should not be a problem if it falls into enemy hands.)

Principle of modern crypto

1. Formal definition
   - System framework (model)：系統長什麼樣子
   - Security definition：如何定義安全
2. Precise assumption Π'
   通常會是已知難題
   從上一節的重點可以知道，我們通常會將加密法與某個已經被研究過的難題 (well-studied hardness) 做連結。若難題不是 well-studied，一來無法說服別人這個加密法安全，二來代表可能有人知道這個問題如何解決。
3. Construction Π
   加密法的步驟是什麼
4. Security proof
   基本上就是上一節的 reduction
   如果假象的攻擊者可以在 definition (即第一個要素) 底下破解 Π，那麼我可以構造另一個攻擊者，使其破解已知難題 Π'。
   上面逆否命題的推論可以寫成：如果 Π' 是安全的 (意即不被破解)，那麼 Π 就是安全的。

加密系統 = 產生 key (key generation) + 加密 (encryption) + 解密 (decryption)

1.7 History of ccryptography

§ Shift cipher

使用 private key encryption。

Key 是每個字母需要做 shift 的次數。

Key generation：選擇一個 $key \in \{0, 1, \ldots, 25\}$
Encryption：將每個字母對應的數字 shift key 位
Decryption：將每個字母對應的數字**反方向** shift key 位

破解：最多嘗試 26 次就可以找到答案

§ Substitution cipher

使用 private key encryption。

Key generation：將每個字母逐一對應到另一個字母，以此這個 mapping 作為 key
Encryption：將明文中的字母按照 key 逐一對應過去 Decryption：將密文中的字母按照 key 逐一對應回來

破解：字典攻擊（常用詞）+ 頻率分析（「E」在英文中出現的次數比較多）

加強：明文中不使用頻率較高的字母

§ Stronger cipher?

Vigenère cipher：設定偏移量為字母在明文中所在的位置。

DES (first published in 1975, and standardized in 1977)

AES

§ History about PKC

1974: Merkle proposed the notion
1976: Diffie-Hellman proposed the key exchange solution (Turing Awad 2015)
1977: Rivest-Shamir-Adleman proposed the first PKE (Turing Award 2002)
UK claimed their Government Communications Headquarters proposed such PKC idea before them.

Other impovements: ID-based encryption from Weil Pairing
使用了不同的 assumption，所以概念上較簡單，執行起來也較有效率（關於 ID-based 的概念，之後如果有時間，可能會提到）