Cryptography and Network Security HW1

B03902033 陳雋

---

1. CIA:
   - Confidentiality: 保護訊息不會被偷看和公佈
     Ex: 保護使用者在通訊軟體中的對話不被外人側錄
   - Integrity: 保護訊息不被竄改
     Ex: 保護使用者在通訊軟體中傳送的對話不被別人偷偷改動
   - Availability: 確保使用者可以使用特定服務
     Ex: 確保通訊軟體提供服務的伺服器不因壞人攻擊而癱瘓

2. Hash:
   - One-wayness: 給定一個 y，很難找到 x 滿足 H(x) = y
     Ex: One-way hash chain 在 Prover/Verifier 情況的應用
   - Weak-collision resistance: 給定一個 x，很難找到 x' 滿足H(x) = H(x')
     Ex: 下載檔案下來的檔案，可以透過 Hash 來確認下載下來檔案是提供者所提供 ，未經竄改(保有完整性)
   - Strong-collision resistance: 難找到 x 跟 x' 滿足 H(x) = H(x')
     Ex: 可讓承諾者不能偷偷更換先前做出的承諾，例如線上剪刀石頭布的例子，承諾者很難偷偷更換出拳。

3. Symmetric Cryptography with KDC
   1. N_A 的功能是防止假伺服器的 Replay Attack，若沒有 N_A 的話，壞人可以利用重播以前的 KDC 傳給 A 的訊息，誘使 A 跟 B 使用舊的金鑰溝通，進一步破解訊息。N_B 的用途則是確認 A 的確持有公用金鑰 K_s 並且還活著。

   2. 攻擊者仍然可以透過 Replay Attack 攻擊這個協議。如果攻擊者算出了某一次對話中的共用金鑰 K_s 和對應的 $E_{Ksb}(K_s \| ID_A)$，他可以透過重播 $E_{Ksb}(K_s \| ID_A)$ 給 B，假扮成 A 建立對話，而 B 沒有辦法判斷這則訊息是否是新的。

   3. 在和KDC對話前，A 先要求和 B 對話
      A -> B: ID_A

      B 給 A 一個以 K_{sb} 加密的 nonce
      B -> A: $E_{Ksb}(ID_A \| N'_B)$

      A 跟 KDC 講說他要跟 B 連線
      A -> KDC: $ID_A , ID_B , N_A , E_{Ksb}(ID_A \| N'_B)$

      KDC 解開 $E_{Ksb}(ID_A \| N'_B)$ 並加入K_s 再加密後，連同 K_s 寄回給 A
      KDC -> A: $E_{Ksa} (N_A \| K_s \| ID_B \| E_{Ksb}(K_s \| ID_A \| N'_B))$

接著 A 就把 $E_{Ksb}(K_s || ID_A || N'_B)$ 傳給 B，之後步驟相同，這裡多出來的 N'_B 即拿來防止第二小題所描述的 Replay Attack。

4. Classical cipher:

R1: 一般的凱薩加密，算出字元差距之後即可重建答案。
R2: 嘗試 26 種字元差距，答案應為其中單字正確數最多的句子
R3: 對不同位置的字元進行不同變換，再以辭典驗證
R4: 跟R3很像，增加找尋金鑰規律的策略，再以辭典驗證
R5: 解出位置之間的對應關係後，找出原來密文
R6: 加密方式是將每隔一段間隔的字元排在一起，找出此距離後即可還原密文
R7: 將密文用 base64 還原
BALSN{C14$5ic41_c!ph3r_1$_r34lly_cl455ic41}

5. Google can beat this:

取出 google 給的兩個 pdf 檔的前面 340 個 byte，之後在後面加上相同的字串即可造出新的 collision。
P1: 用其中一個 pdf 的前面 340 byte，加上任意字串用暴力法找到經過 sha1 計算後最後 24 bit 與題目相符的值
P2: 將 P1 找到的字串附在另一個 pdf 的前 340 byte 後面即可找到 collision
BALSN{D0NT_7RU57_SHA1_N0W}

6. Many-time pad:

運用 Crib dragging 的技巧，猜測句子中可能出現的單字，將兩個句子 XOR 以後用單字在生成句子的各個位置上做 XOR，透過生成單字片段猜測可能單字，進而不斷擴張已知句子長度，直到解出flag。
BALSN{using a key one time is not enough, have you tried using it twice?}

7. Backdoor of Diffie Hellman：

由於 generator 因為 backdoor 而被更改，進而導致 g^a mod p 所生成的循環群變小，因此可以用暴力的方法解出秘密 a，最後再透過找出 g^{ab} mod p 的乘法反元素算出 flag。
BALSN{black magic number}

8. Man in the middle:

先將所有可能的 g mod p 算出來，透過 Reflection Attack 建立兩個連線，在要算密碼的回合兩個連線傳送一個 g mod p，其他回合則交換兩個連線 Server 傳來的訊息回傳，最後若兩個連線所產生的最終密文 XOR 後，跟兩個連線要算密碼的回合傳來訊息之 XOR 相同，即找到了該回合的密碼。有了各個回合的密碼，即可和 Server 建立共同金鑰並解出 flag。
BALSN{Wow_you_are_really_in_the_middle}

9. Only admin can print the flag:

在第一個階段表明 admin 身份，透過 Reflection Attack 由另一個連線得到 Server 要求包含 Nonce 的 Hash 值，接著再透過 Length Extension Attack 在原有的 Hash 值後面添加 printflag 的命令，讓 Server 印出 flag 的值 BALSN{LLLLLLLENTH_EXTENSION_ATTACKKKKKKKKKK!!!}