CNS HW1 REPORT

Problem 1

Confidentially

保密性，確保中間人就算攔截到也無法得知明文。
example : 軍事行動上，不會希望機密資訊被敵軍得知。

Integrity

完整性，確保資訊的完整。
example : 在下載軟體時，我們會希望軟件沒有被添加任何後門或病毒，因此會使用md5之類的hash function，確定完整性。

Availability

可用性，希望目標使用者，能獲得資訊或使用服務。
example : 在現實生活中，惡意人士能透過ddos癱瘓目標服務。

Problem 2

One-wayness

我們會希望hash function是不可逆的，否則有可能因為hash function而被推出系統原始碼等等。

Weak collision resistance

定義 : 給定一個ms，很難找出一個ms1讓hash(ms) = hash(ms1)
我們會希望hash function出的值不容易產生碰撞，如果容易產生碰撞，那將會讓有心人士放後門等，如同現在md5的處境。

Strong collision resistance

定義 : 很難找到一組(x,x') 滿足 x' != x 但是 hash(x) = hash(x')
實際上，因為這個條件過於嚴格，所以根據生日悖論被找到的機率並不低。現實生活中，如果滿足strong collision，可以根據此hash function做身分上的驗證也可做軟體上的驗證。

Problem 3

Part a

因為A有可能送很多Request給KDC，N_A可以讓A確定這個回覆的時間性。如果沒有N_A，攻擊者B，假設他已經畢業了，他可以將以前legal的$E_{K_{SA}}(K_S||ID_B||E_{K_{SB}}(K_S||ID_A))$直接回傳給A，讓他與之產生連線。
N_B讓B確定A是活著的，而且他正在持有這個金鑰。

Part b

一個使用者A可以保留第二階段對於某位使用者B的$E_{K_{SB}}(K_S||ID_A)$直到他畢業，但利用這個，他還是可以跟B通訊。

Part c

我們可以在最一開始，讓A先送一個request給B，然後B會回一個他用語KDC的共用密鑰加密的封包，內部包含一個Nonce跟ID_A，然後再讓A送給KDC，KDC因此可以判斷A是否是legal。

Problem 4

Stage 1 vigenere cipher
Stage 2 凱薩加密 猜平移
Stage 3 相同位子字元的差
Stage 4 加密的差隨著位子遞增

Stage 5 相同字元代表相對位置
Stage 6 柵欄密碼
Stage 7 base64

Problem 5

Google的兩份pdf給出了在預設的IV下，構造collision的結構，所以我們只要先用random戳出一組解，然後再分別透過google的兩個prefix送出即可。

Problem 6

因為所有的可能只有英文字、標點符號跟{}，因此可以先爆搜篩出可能的結果，並且發現倒數第二行有點像flag，然後再根據每個詞填空，做出解答。

Problem 7

因為*gbackdoor*是被除過691829，所以整個group的大小被限制過，只要硬爆691829種，即可找到答案。

Problem 8

Solution 1

如果剛好賽對g，那麼把它給的cipher xor 各個收到的b的hash decode後會剛好是答案，因此searching space是 20^3。

Solution 2

如果開兩個connection，只有一個位子是送自己的猜測，另外兩個位子是互送的話，將兩個flag cipher xor在一起等於k。如果剛好猜對g的話，k xor hash(b) xor hash(b1)會等於0，因此我們只要 3*20次就可以找到各個位子實際上的g。最後再送一輪g給server，然後把server給的B和flag cipher xor完decode，即是解答。

Problem 9

關鍵點是[-1]，所以我們可以根據merkle–Damgård的特性，構造出新的string跟他的sha256。而前面對於nonce的處理，只要用reflection attack即可。