Paper: Juels and R. Rivest. "Honeywords: Making password-cracking detectable." In ACM SIGSAC conference on Computer & communications security, 2013

心得:

這篇論文作者使用 honeywords 來提高 hashed passwords 的安全性，讓攻擊者在嘗試登入提交 honeywords 時會有警示。我很少去想該如何防範攻擊者來竊取密碼的事，透過這篇論文，讓我知道原來在 2013 年就已經有人提出偽密碼的概念(或許更早只是我不知道)，並且提出多種實作想法，我覺得相當有趣。

在密碼學中有"salt"之詞表示在 hash 之前將內容中加入字串，使 hash 結果與加 salt 之前不同，但還是有可能被破解，因此作者也提出了他們的想法如何去增強密碼，像是 chaffing-by-tweaking、chaffing-with-a-password-model、take-a-tail 等等。且將不同使用者的密碼及 honeywords 存放至 honeychecker 中，不過在論文中提及的六種常見的攻擊場景之一: Stolen files of password hashes，描述說攻擊者能夠普遍地竊取許多系統上的密碼哈希文件，或者在不同時間竊取一個系統上的密碼哈希文件。我在想，既然這麼容易被盜取走的話，那作者所提出的 honeychecker 會不會也無法安全地存儲訊息。在第七章中作者考慮了多種攻擊狀況，是關於在論文所提出方法的各種攻擊，有考慮到 honeychecker，不過並沒有描述過多，感覺如何去設計嚴謹的驗證方式也是一個難題。

我實作了論文中提及的方法，chaffing-by-tweaking 及 take-a-tail，前者又分為 chaffing-by-tail-tweaking 及 chaffing-by-tweaking-digits 兩項，在實作時要調整的位置應該僅根據密碼之字符類型的放置模式來選擇位置，而不是選擇特定的字符位置之後再來放置，因為有些人是用單詞作為密碼的一部分，加密後攻擊者還是可以直接認出密碼，所以這三種方式的密碼強度偏低，實作的難度也不高。而 Chaffing-with-a-password-model 的方法，我覺得跟前面的做法相較之下更好，只是需要有足夠多的字詞量去替換，如果在不同密碼中，出現一樣的替換詞次數太多，也會很容易被辨識出來。

不同的方法各有優缺點，此篇論文講述的是使用 honeywords 來保護密碼哈希文件的一些初步嘗試，最後的 Open Problems 中也提到還有很多未解問題，雖然不知道到現在是否已經有人完整的實作整個系統，但是我相信正確的應用 honeywords 的概念會是相當有用的。

```
Password is @310555004_Stu
<Chaffing_by_tail_tweaking>
['@310555004_Stu', '@310555021.Mir', '@310555099<Jtw', '@310555047>Pcw', '@310555047^Fwv', '@310555023_Mel']
<Chaffing_by_tweaking_digits>
['@310907153_Stu', '@310085258_Stu', '@310555004_Stu', '@310533242_Stu', '@310162269_Stu', '@310536125_Stu']

Password is stu_310555004
<Take_a_tail>
['stu_310555994', 'stu_310555036', 'stu_310555004', 'stu_310555449', 'stu_310555444', 'stu_310555822']
```