Honeyword Generation: using some tweaking mechanism

In the first cases, because we don't have the list of RockYou passwords, we utilize some tweaking method mentioned in (1),(2) .

Tail-tweaking

In this method, we tweak the last n characters of the password. The characters are replace by which belong to the same category, i.e.: letters by letters, digits by digits, and special characters by special characters. For example, if we input **"password2016"** and $n = 5$ we might get: **"password2016", "passwore7920", "passworK5784", etc...**

Tweaking-digits

(2) In this case, we tweak the last n digits in the password. We deal with one special case in this method. If the password contain "19XX" and "20XX", we will randomly generate a year in 1900, today.year, so the decoys have more similar characteristic with original password. For example, without this special case, **"cornell2016"** might be transformed to **"cornell8923", "cornell5631", "cornell7314"**, which are less likely generated by normal people. But with this scheme we got **"cornell2014", "cornell1987", "cornell1911"** which are closer to human sense.

Some hand-crafted tweaking mentioned in

(2) In this method, many hand-craft are used.
For number and special characters: Duplication, deletion, number-special key transformation($1- >!$, $\%- > 5$)
For letters: Capitalization, Lower-case, and leet transformation.

References

A. Juels and R. L. Rivest, "Honeywords: Making password-cracking detectable," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pp. 145–160, ACM, 2013.

Y. Zhang, F. Monrose, and M. K. Reiter, "The security of modern password expiration: an algorithmic framework and empirical analysis," in *Proceedings of the 17th ACM conference on Computer and communications security*, pp. 176–186, ACM, 2010.