Créer de tête de nombreux mots de passes inviolables et inoubliables

Nicolas K. Blanchard[1] et Leila Gabasova[2] et Ted Selker[3] et Eli Sennesh[4]

[1] *IRIF, Université Paris Diderot,* [2] *Institut de Planétologie et d'Astrophysique de Grenoble,* [3] *University of California, Berkeley,* [4] *Northeastern University*

Nous présentons un générateur de mots de passe nommé Cue-Pin-Select qui est sécurisé, durable, adaptable à tous les ensembles de contraintes usuelles et aisément exécutable de tête ou avec un papier et un stylo.

Ce générateur extrait de manière pseudo-aléatoire une suites de caractères à partir d'une phrase facile à mémoriser, d'indices locaux et d'un PIN à quatre chiffres. Les mots de passes sont indépendamment sécurisés, et résistent même lorsqu'un adversaire obtient un ou plusieurs mots de passes déjà créés par le générateur.

Mots-clefs : Générateurs de mots de passe, Authentification, Sécurité accessible

1 Introduction

Quel que soit leur niveau de compétence, tous les internautes ont besoin d'une multitude de mots de passe pour leurs différents identifiants virtuels, menant à des vulnérabilités quand ils utilisent le même mot de passe plusieurs fois [GF06]. De plus, les différents services ont souvent des contraintes fortes sur les mots de passe acceptables, privilégiant par erreur la complexité par caractère à la longueur [SKD+14, KSK+11]. Utiliser un mot de passe différent par service nécessite donc des gestionnaires dédiés, ou des mots de passes très simples, ce qui mène à d'autres vulnérabilités [DBC+14, Lip]. Les travaux récents [HB01, BBD13] ont proposé des alternatives mais celles-ci nécessitent ou bien un travail de mémorisation [BV15], ou bien l'aide d'outils numériques pour faire des calculs [SCL12, BBDV14].

Inspirés par les travaux de Blum et al., nous introduisons un système nommé Cue-Pin-Select (ainsi que plusieurs variantes) qui répond aux contraintes suivantes :

— Après une petite période d'adaptation, il devient presque impossible de perdre ses mots de passe.
— Créer un mot de passe prend moins d'une minute, tout comme le recréer si on l'oublie.
— Le système n'a besoin d'aucune machine (ou moyen de stockage) et peut être utilisé de tête.
— Chacun des mots de passes créés a une haute entropie (ici 52 bits).
— Obtenir un des mots de passe en clair (et généralement plusieurs) ne permet pas de retrouver les autres même avec une grande puissance de calcul.

2 Protocole

Le protocole Cue-Pin-Select utilise deux éléments distincts d'informations.

Le premier est une phrase de passe, c'est-à-dire une suite de mots facile à mémoriser et ayant une très haute entropie. Cette phrase s'accompagne aussi d'un code PIN de quatre chiffres, et ces deux secrets sont les seuls qui doivent être conservés en toute sécurité dans la mémoire de l'utilisateur.

Le deuxième élément est un ensemble d'indices ("Cue"), correspondant à une chaîne de quatre caractères associée à un service. Par exemple, Facebook peut devenir FCBK, BOOK, FACE, FABO. Ces indices n'ont pas à être secrets, mais ils doivent être faciles à retrouver (en faisant quelques erreurs au pire).

Avec ces éléments, une utilisatrice peut lancer l'Algorithme 1, dont le fonctionnement est illustré sur la Figure 1.

2.1 Extraction du mot de passe

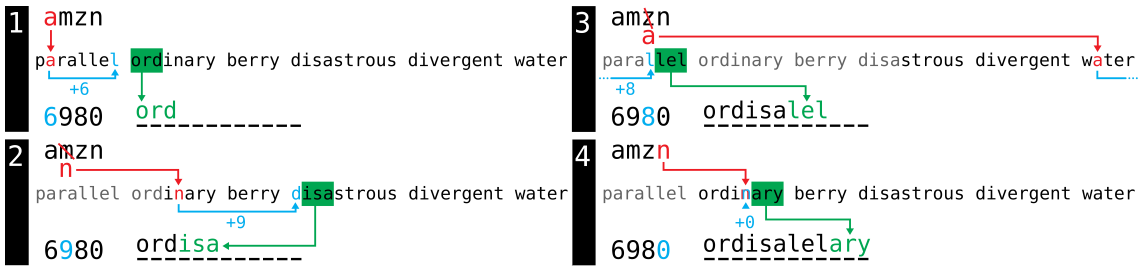
Pour pouvoir extraire de l'entropie d'une source unique de manière déterministe sans trop révéler la source, il est nécessaire d'avoir une phrase ayant beaucoup d'entropie. Pour éviter les phrases trop faciles, nous avons compilé un dictionnaire, correspondant à l'intersection de la liste des 100000 n-grams les plus fréquents compilée par Peter Norvig avec le SOWPODS (liste des mots admissibles au scrabble anglophone) pour obtenir une liste de 87691 mots corrects et fréquents [†]. Chaque phrase est construite par une utilisatrice à partir de 6 mots choisis au hasard dans ce dictionnaire, plus les connecteurs – ou autres mots – de son choix.

Données: Phrase de passe P d'au moins six mots ; PIN K de quatre chiffres ; nom du service N
Sortie: String S de 12 caractères alphabétiques

Début
  Créer à partir de N un string M de quatre caractères facile à mémoriser
  $L \longleftarrow$ Longueur(P), $V \longleftarrow 0$, $S \longleftarrow$ ""
  pour $i = 0\,;\ i < 4\,;\ i{+}{+}$ **faire**
    $X \longleftarrow M[i]$
    tant que $X \notin P$ **faire**
      $X \longleftarrow$ lettre succédant X dans l'alphabet
    $V \longleftarrow$ indice de la prochaine occurrence de $X \in P$ après V (mod L)
    $V \longleftarrow V + K[i] + 3 \pmod{L}$
    $S \longleftarrow$ Concaténer $(S, P[V-2], P[V-1], P[V])$
    `/* Tous les indices sont modulo L */`
  Renvoyer S

Algorithm 1: Cue-Pin-Select

FIGURE 1: Les quatre phases de Cue-Pin-Select



3 Sécurité

Une hypothèse utilisée dans la preuve de sécurité est que chaque triplet de lettres est tiré selon une bonne distribution (proche de celle correspondant à un triplet provenant d'un corpus de texte). Cette hypothèse repose entièrement sur la présence du code PIN, car sans cette phase de l'algorithme, lire la lettre 'Q' donnerait lieu à 2.7 bits d'entropie pour le triplet suivant, alors que 'o' en donnerait 10.4. Le décalage supplémentaire apporté par le PIN permet donc d'éviter tout cela, tout en augmentant la distance parcourue ce qui permet d'éviter qu'un adversaire connaisse la position réciproque de deux trigrammes (nous avons montré avec des simulations que le premier est souvent avant le deuxième, mais qu'aucune différence statistique notable n'est présente pour les suivants).

†. Ce dictionnaire peut paraître étendu, mais il contient de nombreuses formes de chaque mot (conjugaisons et pluriel notamment), et des études ont montré qu'un étudiant américain moyen connaît plus de 200000 telles formes.

3.1 Attaques par force-brute et par dictionnaire

Un adversaire voulant trouver la phrase de passe par force brute doit faire face à un minimum de 210 bits d'entropie. Cette mesure n'est cependant pas adaptée à ces considérations, et nous devons analyser l'entropie de la phrase en partant du principe que l'adversaire connaît parfaitement l'algorithme. Dans ce cas, nous arrivons à un minimum de 6 mots choisis uniformément parmi 87691, plus un PIN, menant à un total de 111 bits. Les attaques par dictionnaire ne sont donc pas non plus envisageables.

Pour les mots de passe, grâce au dictionnaire étendu, nous avons une haute entropie par lettre. En effet, nous avons calculé l'entropie des trigrammes (en faisant l'hypothèse d'uniformité mentionnée précédemment), et si elle varie selon leur position entre 11.4 et 14 bits (respectivement pour suffixes et préfixes avec les infixes à 13 bits), nous atteignons en moyenne 52 bits d'entropie, remarquablement proche du maximum théorique de 56 bits pour un mot de passe alphabétique de 12 caractères.

3.2 Attaques avec mots de passes en clair

Afin de simplifier le modèle d'attaque et de donner une borne inférieure sur la sécurité, nous supposons qu'un adversaire ne dispose pas uniquement du mot de passe en clair, mais aussi de la position de chacun des trigrammes dans la phrase de passe originale (et par conséquent, de la longueur de cette phrase).

Nous avons mené 10^4 simulations, en créant à chaque fois un couple (phrase/mot de passe) puis en calculant le nombre de possibilités restantes en ayant les lettres du mot de passe révélées. La courbe sur la Figure 2 montre que nous pouvons garantir une entropie supérieure à 36 bits dans le pire cas (et de 54 bits en moyenne). Contre des attaques avec plusieurs mots de passe en clair, la sécurité baisse mais reste élevée comme vu sur la Figure 3 (avec 5×10^3 simulations à chaque fois, donnant en moyenne respectivement 32 et 20 bits avec 2 et 3 mots de passes révélés). Les cas avec très peu d'entropie sont dus au fait que la phrase de passe peut être de longueur inférieure à 33 (avec probabilité 0.5%), et que nous révélons potentiellement jusqu'à 36 lettres avec trois mots de passe.

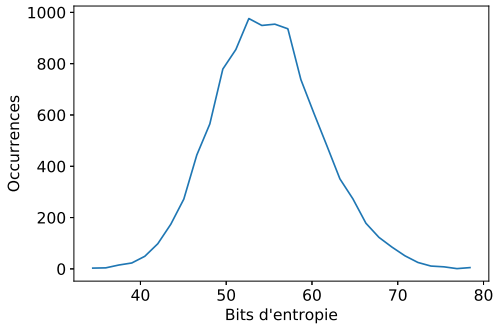


FIGURE 2: Entropie avec un mot de passe

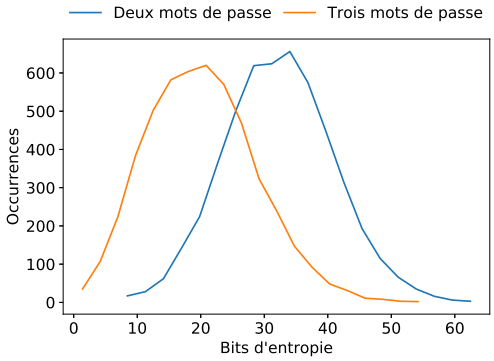


FIGURE 3: Entropie avec deux et trois mots de passe

4 Ergonomie

Afin de vérifier l'ergonomie du système, nous avons aussi fait appel à 11 alpha-testeurs non spécialistes, dont l'âge variait entre 17 et 65 ans. Chacun devait créer ou recréer deux ou trois mots de passes deux fois par jour pendant quatre jours.

4.1 Mémoire

La plupart des internautes étant poussés à créer très régulièrement de nouveaux mots de passe, l'utilisation fréquente de l'algorithme devrait non seulement augmenter la vitesse mais aussi empêcher les utilisateurs d'oublier leur phrase de passe. En pratique, la répétition régulière de la phrase la rend en effet presque inoubliable. Ainsi, le deuxième jour, seul un alpha-testeur avait du mal à s'en souvenir (un des six mots lui manquait), et après cela aucun n'eut de problème.

4.2 Effort

Cette contrainte est la plus importante car un système trop coûteux ne sera pas utilisé, ou bien ses mécanismes de sécurité seront contournés. Ici, le coût correspond au temps nécessaire et à la difficulté d'exécution de l'algorithme. Pour la difficulté de l'algorithme, plusieurs testeurs eurent du mal à créer leurs premiers mots de passe (prenant du temps et faisant des erreurs). Heureusement, si ce coût fut initialement élevé (certains testeurs prenant trois à quatre minutes pour créer un mot de passe avec papier et stylo), l'apprentissage se fit très vite. Au matin du troisième jour, la médiane et la moyenne tombèrent à 40s et 42s, avec des temps entre 30s et 52s. Après une journée et demi d'entraînement supplémentaire cette fois sans papier ni stylo (calcul mental uniquement), les temps passés (qui étaient remontés à plusieurs minutes) redescendirent à 54s et 57s, pour des temps allant de 24s à 71s.

4.3 Adaptabilité

Un problème majeur pour la durabilité d'un générateur de mots de passe est qu'il doit être capable de gérer toutes les contraintes imposées par les fournisseurs de service. Sans cela, il ne peut suffire, et force l'utilisateur à changer ses règles ou à utiliser plusieurs générateurs en parallèle, réduisant fortement leur intérêt. Nous avons développé une extension simple et rapide (mais qui ne tient pas ici) rendant Cue-Pin-Select adaptable et capable de satisfaire – de manière déterministe – toutes les contraintes fréquemment rencontrées, y compris celles qui forcent l'utilisateur à changer de mot de passe tous les mois sans répétition.

5 Conclusion

Nous avons introduit le système Cue-Pin-Select qui est hautement résistant à divers types d'attaques tout en restant utilisable dans la vie de tous les jours grâce à une forte amélioration des performances au cours des premiers jours observée chez tous les alpha-testeurs. Les bornes inférieures de sécurité données reposent sur un modèle d'adversaire fort et il serait intéressant d'avoir une analyse plus poussée correspondant à une situation réaliste. Avec une telle sécurité, et même si certains utilisateurs avaient franchi la barre des 30s, il semble qu'un meilleur compromis pourrait être trouvé entre sécurité et simplicité de l'algorithme. Cependant, toutes les variantes simplifiées de Cue-Pin-Select explorées jusqu'ici souffrent de sérieuse failles de sécurité.

Références

[BBD13] Jeremiah Blocki, Manuel Blum, and Anupam Datta. Naturally rehearsing passwords. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 361–380. Springer, 2013.

[BBDV14] Jeremiah Blocki, Manuel Blum, Anupam Datta, and Santosh Vempala. Towards human computable passwords. *arXiv preprint arXiv :1404.0024*, 2014.

[BV15] Manuel Blum and Santosh Srinivas Vempala. Publishable humanly usable secure password creation schemas. In *Third AAAI Conference on Human Computation and Crowdsourcing*, 2015.

[DBC+14] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. The tangled web of password reuse. In *NDSS*, volume 14, pages 23–26, 2014.

[GF06] Shirley Gaw and Edward W. Felten. Password management strategies for online accounts. In *Proceedings of the Second Symposium on Usable Privacy and Security*, SOUPS '06, pages 44–55, New York, NY, USA, 2006. ACM.

[HB01] Nicholas J Hopper and Manuel Blum. Secure human identification protocols. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 52–66. Springer, 2001.

[KSK+11] Saranga Komanduri, Richard Shay, Patrick Gage Kelley, Michelle L. Mazurek, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Serge Egelman. Of passwords and people : Measuring the effect of password-composition policies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2595–2604, New York, NY, USA, 2011. ACM.

[Lip] Peter Lipa. The security risks of using 'forgot my password' to manage passwords. `https://www.stickypassword.com/blog/the-security-risks-of-using-forgot-my-password-to-manage-passwords/`. Accessed : 2017-12-18.

[SCL12] Hung-Min Sun, Yao-Hsin Chen, and Yue-Hsun Lin. opass : A user authentication protocol resistant to password stealing and password reuse attacks. *IEEE Transactions on Information Forensics and Security*, 7(2) :651–663, 2012.

[SKD+14] Richard Shay, Saranga Komanduri, Adam L. Durity, Phillip (Seyoung) Huh, Michelle L. Mazurek, Sean M. Segreti, Blase Ur, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Can long passwords be secure and usable ? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 2927–2936, New York, NY, USA, 2014. ACM.