Créer de tête de nombreux mots de passe inviolables et inoubliables

Nicolas K. Blanchard [1], Leila Gabasova [2], Ted Selker[3], 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

20èmes Rencontres Francophones pour les Aspects Algorithmiques des Télécommunications 31 mai 2018

Comment se débarrasser du problème de réutilisation des mots de passe ?

Trop de mots de passe

État actuel de l'utilisation des mots de passe :

- Un individu moyen doit se souvenir d'une centaine de mots de passe
- Création de 50 mots de passe par an en moyenne
- Nombreuses contraintes contre-productives sur les types de caractères (1@MyPassword) et changements réguliers obligatoires

Trop de mots de passe

État actuel de l'utilisation des mots de passe :

- Un individu moyen doit se souvenir d'une centaine de mots de passe
- Création de 50 mots de passe par an en moyenne
- Nombreuses contraintes contre-productives sur les types de caractères (1@MyPassword) et changements réguliers obligatoires

D'où :

- Réutilisation massive (75% des utilisateurs)
- Partage fréquent (40% des utilisateurs)
- Perte et réinitialisation fréquente (entre 40% et 60% des mots de passes importants sont réinitialisés tous les 3 mois)

Solutions proposées

Solutions externes proposées et dangers:

- Méthodes biométriques (vulnérable au hack et inchangeables)
- Facebook Connect et assimilés (confiance en un tiers)
- Gestionnaire de mots de passe (unique point de défaillance)
- Post-it ou aide-mémoire physique (utilisés par 49% malgré le risque de vol)

Solutions proposées

Solutions externes proposées et dangers:

- Méthodes biométriques (vulnérable au hack et inchangeables)
- Facebook Connect et assimilés (confiance en un tiers)
- Gestionnaire de mots de passe (unique point de défaillance)
- Post-it ou aide-mémoire physique (utilisés par 49% malgré le risque de vol)

Solutions internes proposées et limites:

- Mot de passe avec sel variable (bientôt vulnérable)
- Méthode de Blum (coût élevé)

Contraintes

Contraintes pour une création de mot de passe durable et utilisable :

- Haute entropie pour chaque mot de passe
- Haute entropie même contre un vol de mot de passe en clair

Contraintes

Contraintes pour une création de mot de passe durable et utilisable :

- Haute entropie pour chaque mot de passe
- Haute entropie même contre un vol de mot de passe en clair
- Mémorisable même sans utilisation d'un mot de passe pendant longtemps (et donc déterministe)
- Compréhensible et utilisable rapidement par un humain moyen

Contraintes

Contraintes pour une création de mot de passe durable et utilisable :

- Haute entropie pour chaque mot de passe
- Haute entropie même contre un vol de mot de passe en clair
- Mémorisable même sans utilisation d'un mot de passe pendant longtemps (et donc déterministe)
- Compréhensible et utilisable rapidement par un humain moyen
- Compatible avec les contraintes de divers services

Idée : avoir un grand secret dont on extrait de l'entropie de tête

Cue-Pin-Select

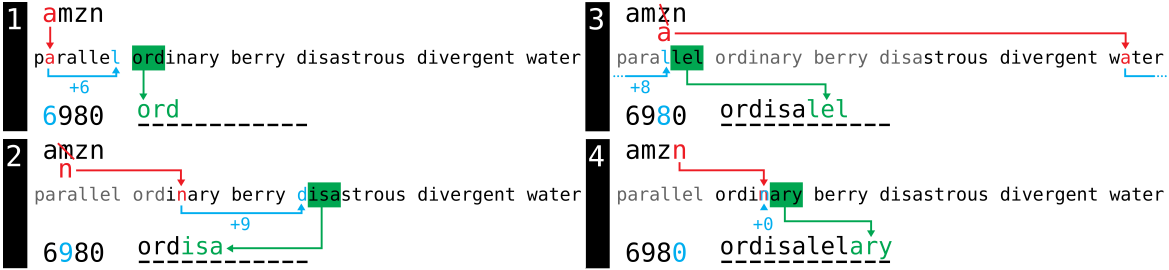
Principe de la méthode :

- Créer une phrase de passe à haute entropie et un PIN de 4 chiffres
- Pour chaque service, générer un code à 4 lettres
- Extraire de la phrase 4 triplets de caractères de manière déterministe

Algorithme initial

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

Exemple appliqué



Attaques en force brute et par dictionnaire

Recommandation pour services web : 36-42 bits généralement (30 ans à 1000 essais par seconde).

Force brute contre Cue-Pin-Select :

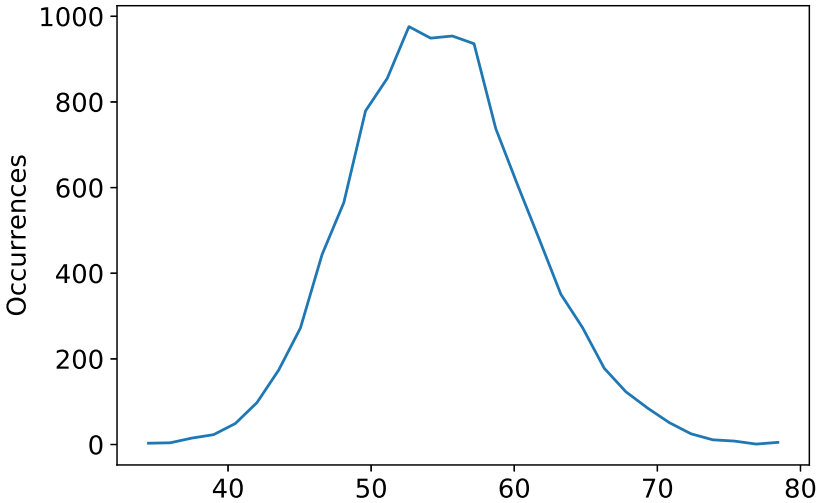
- Contre mot de passe $\rightarrow$ 56 bits
- Avec dictionnaire probabiliste contre mot de passe $\rightarrow$ 52 bits
- Contre phrase de passe $\rightarrow$ 210 bits
- Avec dictionnaire contre phrase $\rightarrow$ 111 bits

Attaques avec mot de passe en clair

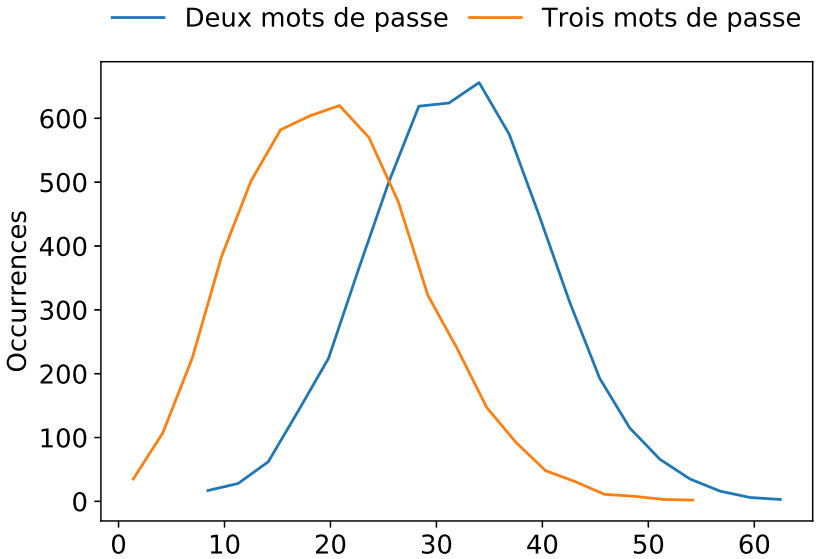
Modèle d'adversaire puissant connaissant :

- Un ou plusieurs mots de passe
- La longueur de la phrase
- La position des différents triplets connus dans la phrase

Entropie mots de passe en clair (empirique sur 10 000 essais)



Entropie mots de passe en clair (empirique sur 10 000 essais)



Protocole de l'expérience utilisateur

Test de la méthode sur 11 personnes ayant des âges et des parcours différents.

Pendant 4 jours:

- Création de 2-3 mots de passe matin et soir (parfois régénération au lieu de création)
- Avec papier et stylo, phrase devant soi (et correction d'erreurs) le premier jour
- Sans phrase le deuxième jour
- Sans aide papier à partir de la seconde moitié du troisième jour

Résultats de l'expérience utilisateur

Temps pris le premier jour

Temps(s)	Tâche 1.1a	1.1b	1.1c	1.1d	1.2a	1.2b
Moyenne	89	82	72	63	70	59
Médiane	72	56	51	56	66	55
Max	233	211	222	108	132	113
Min	47	35	35	32	32	33

Temps pris le deuxième jour et troisième jour au matin

Temps(s)	Tâche 2.1a	2.1b	2.2a	2.2b	3.1a	3.1b
Moyenne	50	49	54	45	51	42
Médiane	44	47	51	40	50	40
Max	87	68	70	61	74	53
Min	30	32	42	31	38	30

Temps pris le troisième après-midi et le quatrième jour

Temps(s)	Tâche 3.2a	3.2b	4.1a	4.1b	4.2a	4.2b	4.2c
Moyenne	105	86	81	74	67	58	57
Médiane	90	80	77	71	65	56	54
Max	220	131	130	117	106	86	71
Min	65	47	46	47	24	33	31

Adaptabilité

Algorithme étendu pour gérer les cas suivants:

- Numéros et caractères spéciaux : rajout standard au milieu du mot de passe

Adaptabilité

Algorithme étendu pour gérer les cas suivants:

- Numéros et caractères spéciaux : rajout standard au milieu du mot de passe
- Limite de taille : ajout standard ou mot de passe tronqué

Adaptabilité

Algorithme étendu pour gérer les cas suivants:

- Numéros et caractères spéciaux : rajout standard au milieu du mot de passe
- Limite de taille : ajout standard ou mot de passe tronqué
- Changement fréquents : offset sur le pin/cue

Résultats

Cue-Pin-Select:

- Sécurité à 52 bits pour chaque mot de passe
- Résistance à un mot de passe perdu minimum, généralement deux, même sous des hypothèses fortes
- Possibilité de créer plus de 500 mots de passe sécurisés sans recoupement
- Exécution rapide avec amélioration grâce à l'apprentissage, temps de calcul inférieur à 1 min
- Mémorisation durable selon les modèles, même sans utilisation régulière
- Adaptable aux contraintes de sécurité les plus courantes
- Extension naturelle améliorant la sécurité

Travail futur

Amélioration de l'analyse :

- Modèles d'attaques avec des hypothèses moins fortes pour améliorer la garantie de sécurité
- Extension à d'autres langues ayant moins de mots

Travail futur

Amélioration de l'analyse :

- Modèles d'attaques avec des hypothèses moins fortes pour améliorer la garantie de sécurité
- Extension à d'autres langues ayant moins de mots

Amélioration de l'algorithme :

- Développement d'un modèle de coût de calcul humain pour simuler performance sur un algorithme sans besoin d'une expérience utilisateur
- Recherche d'un algorithme potentiellement moins résistant mais exécutable en moins de 30s

Merci pour votre attention