Vers un Code Personnel d'Identité Respectueux de la Vie Privée

Denis Migdal, Christophe Rosenberger

Vers un Code Personnel d'Identité Respectueux de la Vie Privée

D. Migdal[1] C. Rosenberger[1]

[1] Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France

{denis.migdal, christophe.rosenberger}@ensicaen.fr

Résumé

De nombreuses applications sur Internet nécessitent d'avoir des informations sur l'internaute, pour vérifier qu'il a bien le droit à d'accéder à un service numérique (vérification d'une preuve d'identité comme un mot de passe), pour éviter des attaques (pédopornographie, usurpation de profil,...) ou pour donner de la confiance aux autres utilisateurs (réseaux sociaux). Nous proposons dans ce papier une méthode de génération d'une signature basée sur l'identité de l'individu. Elle est calculée à partir de 1) la collecte de données biométriques sur l'individu, sur son ordinateur, son navigateur web, 2) le pré-traitement de ces données et 3) la protection des données personnelles pour la génération d'un code binaire. Nous illustrons l'intérêt de la méthodologie proposée avec des résultats préliminaires sur des données personnelles réelles.

Mots clefs

Informations personnelles, biométrie comportementale, protection de données personnelles.

1 Introduction

La consommation de services numériques sur Internet est de nos jours importante que ce soit pour les réseaux sociaux, le commerce électronique ou les jeux en ligne. À titre d'exemple, en 2016, 96% des français ayant demandé un extrait du casier judiciaire l'ont fait sur Internet [1]. Néanmoins, plusieurs données personnelles peuvent être récupérées lors de l'usage d'un service numérique sur Internet soit fournies par l'internaute (notamment sur les réseaux sociaux) soit collectées automatiquement.

Les services numériques sur Internet collectent de plus en plus des données personnelles liées à l'internaute parfois à des fins légitimes (détection de fraude, examen à distance, ...) mais aussi à des fins non conformes aux conditions de collecte (vente à d'autres services, consolidation d'identités, ...). Ces données personnelles peuvent être liées à l'individu (donnée biométrique, nom, âge, ...), au navigateur (version, type, ...), à la machine de l'internaute (système d'exploitation, matériel, résolution de l'écran). Toutes ces informations parfois collectées dans un contexte légitime peuvent aller jusqu'à identifier l'individu ce qui pose un problème majeur de respect de la vie privée.

La principale contribution de ce papier est de proposer une méthode de génération d'un code binaire lié à l'identité numérique d'un individu. Ce code ne permet pas de remonter aux informations utilisées pour le calculer et permet également de réaliser des comparaisons avec d'autres codes. Nous présentons les différentes étapes de calcul de ce code. Les informations utilisées vont du navigateur utilisé, à la machine, jusqu'à l'individu. Des pré-traitements sont réalisés sur ces données afin de calculer le code dans la dernière étape. Nous présentons brièvement quelques applications intéressantes de ce code (authentification ou identification d'attaques comme la détection de plusieurs comptes associés à une identité).

Cet article est organisé comme suit. La section 2 présente un état des travaux antérieurs sur la collecte et l'utilisation de données personnelles. La méthode proposée est décrite dans la section 3. La section 4 présente des résultats préliminaires sur des données réelles. Nous concluons et donnons quelques perspectives dans la section 5.

2 Travaux antérieurs

Le *Browser Fingerprinting* permet de suivre les utilisateurs dans leur navigation Internet grâce aux données discriminantes qu'un service donné peut récolter, souvent dans l'objectif de proposer des "services personnalisés" correspondant au profil-type de l'utilisateur. Les sites Panopticlick [3], IAmUnique [5], et UniqueMachine [2] permettent de calculer son *Browser Fingerprint* à partir des données collectées par le site, généralement via le réseau et l'API JavaScript afin de déterminer le degré d'unicité de l'empreinte calculée parmi celles déjà collectées. Plus le *browser fingerprint* est unique, plus un service aura capacité à le discriminer.

Cependant, le *browser fingerprint* peut varier, e.g. par le changement du navigateur, de sa configuration [6], ou tout simplement de machine. Le but n'est pas d'identifier l'utilisateur de façon certaine, mais d'identifier un ensemble de sessions de navigations appartenant à un même utilisateur. Les données utilisées pour le *browser fingerprinting* peuvent être liées, e.g., au matériel (e.g. carte graphique

1. D'après les chiffres 2017 du SGMAP, https://goo.gl/wNh3kH

[2], écran), au système d'exploitation, au navigateur utilisé, à sa configuration, aux polices installées [3, 5], à l'historique du navigateur [9], ou aux domaines bloqués [1].

3 Méthode proposée

L'objectif de la méthode proposée est de calculer un code binaire lié à une personne à partir d'informations personnelles (techniques et biométriques). Ce code doit répondre à différentes exigences :

- *Non inversible* : le code binaire de l'utilisateur ne doit pas donner d'informations sur les données personnelles collectées.
- *Confidentialité* : la valeur des attributs ne peut être connue, ni déduite, par le service.
- *Conservation de la similarité* : Si les données personnelles d'un individu sont similaires alors les codes binaires résultant doivent l'être.
- *Non-usurpation* : un tiers ne peut forger un code lui permettant d'usurper un utilisateur légitime.
- *Révocation* : l'utilisateur légitime doit pouvoir révoquer un code binaire existant.

Dans le cadre de cet article, un score de confiance peut être calculé avec la distance de Hamming entre la preuve et l'engagement, tous deux vecteurs binaires de taille fixe. Aussi, nous considérerons, et approfondirons les modalités d'informations personnelles suivantes :

- ce que l'utilisateur est/sait faire : sa biométrie comportementale ;
- ce que l'utilisateur possède : son navigateur ;
- où l'utilisateur est : sa localisation physique et organisationnelle ;
- "ce que l'utilisateur préfère" : sa configuration.

La figure 1 présente le principe général de la méthode proposée. Un simple mot de passe est utilisé comme clé secrète [4]. Dans ce cas, Alice par la saisie du mot de passe consent à donner ce code binaire au service. Les différentes étapes de calcul sont présentées par la suite.

3.1 Collecte de données personnelles

A l'heure actuelle, il est possible de collecter un grand nombre de données personnelles. Nous détaillons les informations collectées par grande catégorie.

Navigateur. Afin d'authentifier un navigateur, une simple clé stockée sur ce dernier suffit. La clé, que nous nommerons *localkey*, est une valeur de n bits générée aléatoirement au premier usage du navigateur, utilisée ensuite pour l'authentifier. Pour n suffisamment grand, la probabilité de collision est négligeable, et la recherche exhaustive difficile. Dans le cadre de l'expérience, n=64, pour des besoins en sécurité plus importants, la taille de la clé peut-être augmentée, e.g. n=512.
La clé peut être stockée dans le localStorage [2] du navigateur, ou, idéalement, dans le simple-storage d'une WebExtension. Il est cependant possible à un attaquant de subtiliser la clé s'il a accès à la machine, ou à la session de l'utilisateur. Les clés étant générées aléatoirement, la compromission d'une clé ne compromet pas les clés des autres navigateurs possédés par l'utilisateur. Il est possible de protéger la clé, e.g. en la chiffrant, ainsi que d'en détecter l'utilisation frauduleuse, e.g. via les autres informations personnelles. Cependant, ceci ne sera pas abordé dans le cadre de cet article.

Localisation. Les adresses IP sont distribuées par plages, de l'IANA [3] aux RIR [4], des RIR aux RIL [5], et enfin des RIL aux utilisateurs. Il est ainsi possible d'en déduire le réseau de l'utilisateur, et sa position administrative (e.g. département) ou physique (e.g. position GPS). Cependant, le réseau TOR, un VPN, ou un proxy, peuvent être utilisés pour masquer l'adresse IP de l'utilisateur. Le réseau et positions déduites de l'adresse IP seront alors ceux du proxy, du VPN, ou du nœud TOR sortant.
Dans le cadre de cet article, les localisations administrative (pays, région, département, ville) et physique (latitude et longitude) sont déterminées via l'API Google Map à partir d'une adresse extraite de la base dp-ip [6]. Dans un travail futur, il serait aussi possible de déduire, soit le FAI (Fournisseur d'adresse Internet) de l'utilisateur, soit sa localisation structurelle au sein d'une entité (e.g. entreprise, université, centre de recherche, structures gouvernementales), à l'aide de requêtes DNS, reverse DNS, WHOIS IP, et WHOIS domain. Il est aussi possible d'avoir plus d'informations sur l'adresse IP à l'aide de DNSBL [7].

Données réseau. Les données envoyées au service par les protocoles de communication sont discriminantes et permettent, par des techniques de *browser fingerprinting*, d'identifier l'utilisateur [3, 5]. De manière analogue, ces données peuvent être exploitées pour authentifier l'utilisateur en les comparant avec les données d'enrôlement. Ainsi, cette modalité ne peut être utilisée si les données sont, à chaque échange, générées aléatoirement. Cependant, l'usurpation est triviale pour qui a connaissance de ces données, e.g. pour qui fournit un service à l'utilisateur. De même, l'utilisation de données normalisées, e.g. via l'utilisation du navigateur TOR, augmente la probabilité de collision. Cette modalité accorde ainsi peu de confiance en l'authentification de l'utilisateur, mais permet de détecter la réception de données inhabituelles.
Dans le cadre de cet article, les champs suivants sont extraits de l'en-tête HTTP :

- *User-Agent* : chaîne de caractère arbitraire définie par le navigateur.
- *Accept, Accept-Language, Accept-Encoding* : préférences (valeurs $\in [0, 1]$) quant aux formats, langues, et encodages à utiliser.
- *Referer* : URL de la page précédente, parfois retiré, tronqué, ou aléatoire.

2. fonctionnalité HTML5

3. Internet Assigned Numbers Authority
4. Registres Internet Régionaux
5. Registres Internet Locaux
6. download.dp-ip.com/free/dbip-city-2017-05.csv.gz
7. DNS Blacklist

Figure 1 – *Principe de la méthode proposée*

— *Cookie* : cookies envoyés par le navigateur.
— *DNT, Connection, Upgrade-Insecure-Requests* : autres paramètres.

Données biométriques. La biométrie comportementale de l'utilisateur peut être analysée à partir de ses actions claviers et souris, décrits, dans le navigateur, par les événements JavaScript . Dans le cadre de cet article, la dynamique de frappe de l'utilisateur est représentée à partir des 20 digrammes les plus fréquents : *"r ", "te", "nt", ", ", "n ", "en", " s", "le", " l", " c", "de", ('arrowleft', 'arrowleft'), " p", " d", "on", "t ", "es", "s ", "e ", ('backspace', 'backspace')*. Plus précisément, les durées suivantes seront étudiées :

— P_1R_1 : d'appui du premier caractère.
— P_2R_2 : d'appui du second caractère.
— P_1P_2 : entre les pressions des deux caractères.
— R_1R_2 : entre les relâchements des deux caractères.
— R_1P_2 : entre le relâchement du premier caractère et la pression du second.
— P_1R_2 : entre la pression du premier caractère et le relâchement du second.

3.2 Pré-traitement des données

Afin d'obtenir pour chaque modalité, un vecteur de réels de taille fixe, les données collectées sont converties en vecteurs de réels, puis concaténées. La distance entre deux vecteurs pouvant fortement être influencée par les valeurs extrêmes, ces dernières sont normalisées.

Navigateur. Localkey, clé de n bits, est convertie en un vecteur de n bits. Ainsi, la localkey de 16 bits, *"0x0123"*, est convertie en *[0,0,0,0, 1,0,0,0, 0,1,0,0, 1,1,0,0]*.

Localisation. L'adresse IP est convertie en un vecteur composé :

— d'un vecteur composé des bits de l'adresse IP divisés par 2^{32-p-1} avec p, poids du bit.
— d'un vecteur de bits des $128/2^k$ premiers octets du hash md5 du nom de chaque localité avec k=1 pour "pays", k=2 pour "région", k=3 pour "département", et k=4 pour "ville".
— d'un vecteur de 3 angles $\in [-90; +90]$ représentant la latitude (lat) et la longitude l (lng1, lng2) de la localisation GPS. lng1 et lng2 valent $sign(\alpha) * ||\alpha| - (|\alpha| > 90) * 180|$ avec $\alpha = l$ pour lng1 et $\alpha = rot90(l) = (l - 90)\%360 - 180$ pour lng2. Ces angles sont normalisés par la formule suivante : $(angle + 90)/180$.

Données réseau. *Referer, User-Agent, Connection* et *Cookie* sont convertis en histogrammes, vecteurs donnant pour chaque caractère son effectif. Seuls les caractères ASCII $\in [0x20, 0x7F[$, soit 95 caractères, sont considérés. *Accept, Accept-Encoding,* et *Accept-Language* sont convertis en vecteurs donnant la préférence pour chaque format, encodage, et langue présents dans une liste prédéfinie. Une valeur supplémentaire indique la présence d'espaces après les virgules présentes dans le champ. *DNT* et *Upgrade-Unsecure-Requests* sont convertis en vecteurs de un entier valant 1 si positionné, 0 sinon. Les listes prédéfinies sont :

— Accept : "text/html", "application/xhtml+xml", "application/xml", "image/webp", "image/jxr" ;
— Accept-Encoding : "gzip", "deflate", "br", "sdch" ;
— Accept-Language : "fr", "fr-FR", "en-US", "en".

Données biométriques. Les durées collectées sont converties en un vecteur donnant, pour chaque digraphe considéré, les moyennes des 6 durées. Ces moyennes sont converties en millisecondes, limitées à 1000 puis divisées par 1000.

3.3 Protection des données

L'enjeu que nous souhaitons adresser dans ce travail est la possibilité de répondre à des applications de services numériques sur Internet (authentification, détection d'attaque, ...) tout en préservant le respect de la vie privée de l'individu. À partir des données personnelles collectées, nous

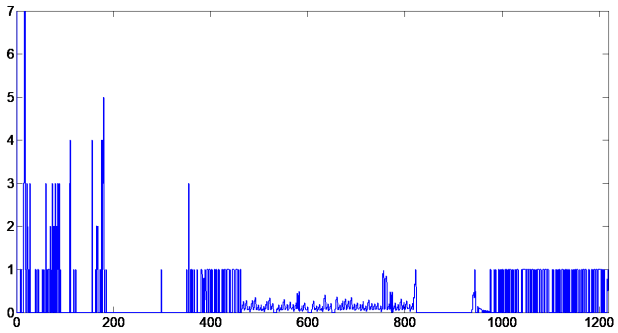


Figure 2 – *Un exemple des valeurs brutes après pré-traitement (1218 nombres réels). En abscisse l'identifiant du nombre, en ordonné sa valeur.*

souhaitons générer une signature binaire comme caractéristique dynamique d'un individu ayant perdu son caractère sémantique. Au final, le service numérique peut exploiter cette signature binaire sans connaître les informations utilisées pour la générer.

L'algorithme Biohashing est un algorithme bien connu dans le domaine de la biométrie. Il permet de transformer des données biométriques représentées par un vecteur à valeur réelle de longueur fixe et génère un modèle binaire appelé BioCode de longueur inférieure ou égale à la taille d'origine. Cette transformation est non inversible et permet de conserver la similarité des données en entrée. Cet algorithme a été initialement proposé pour le visage et les empreintes digitales par Teoh *et al.* dans [8]. L'algorithme de Biohashing est applicable sur toutes les modalités biométriques, voire données personnelles, pouvant être représentées par un vecteur de valeurs réelles de longueur fixe. Cette transformation nécessite un secret liée à l'utilisateur. Dans notre cas, il pourra s'agir d'un mot de passe saisi par l'utilisateur [4]. La comparaison des BioCodes est réalisée par le calcul de la distance de Hamming. L'algorithme de Biohashing transforme un vecteur de paramètres $T = (T_1, \dots T_n)$ dans un modèle binaire appelé BioCode $B = (B_1, \dots B_m)$, avec $m \leq n$, comme suit :

1. m vecteurs aléatoires orthonormés $V_1, \dots, V_m$ de la longueur n sont générés à partir d'un secret servant de germe du tirage aléatoire (typiquement avec l'algorithme de Gram Schmidt).
2. Pour $i = 1, \dots, m$, calcul du produit scalaire $x_i =< T, V_i >$.
3. Calcul du BioCode $B = (B_1, \dots, B_m)$ avec le processus de quantification :

$$B_i = \begin{cases} 0 & \text{if} \quad x_i < \tau \\ 1 & \text{if} \quad x_i \geq \tau, \end{cases}$$

Où τ est un seuil donné, généralement égal à 0.

La performance de cet algorithme est assurée par le produit scalaire avec les vecteurs orthonormés, tels que détaillés dans [7]. Le processus de quantification garanti la non-inversibilité des données (même si $n = m$), car chaque coordonnée de l'entrée T est une valeur réelle, alors que le BioCode B est binaire. Nous proposons d'utiliser cette transformation dans la protection des données personnelles.

4 Expérimentations

Dans cette partie, nous détaillons le protocole expérimental utilisé. Quelques résultats préliminaires sont donnés afin de montrer l'intérêt du calcul du code binaire.

4.1 Protocole expérimental

Une campagne de collecte a été organisée en mars 2017 sur le site trust.greyc.fr. Les participants ont été recrutés via les listes de diffusion du laboratoire GREYC et de l'école d'ingénieur ENSICAEN. De ce fait, les données collectées proviennent de membres sur un lieu assez unique. En effet, la majorité des participants sont localisés à Caen, utilisent les mêmes réseaux (ENSICAEN et UNICAEN), et ont donc la même adresse IP sortante. De plus, l'utilisation des postes des structures du GREYC et de l'ENSICAEN, font que les participants ont des configurations similaires, et ainsi des données réseau proches.

Avec seulement 22 participants, majoritairement localisés sur Caen, l'échantillon n'est pas représentatif, mais permet une première expérimentation du code personnel d'identité. Lors de la collecte, les participants sont invités à répondre à 8 questions relatives à la vie privée, puis à recopier un extrait de la Déclaration Universelle des Droits de l'Homme (voir 3). Afin d'éviter toute influence sur la dynamique de frappe au clavier, les participants ne sont informés de la collecte d'informations qu'à partir de l'étape 5 où ils sont invités à renseigner leurs informations personnelles. Toutes les données collectées sont stockées dans le sessionStorage du navigateur et ne sont soumises qu'après validation de l'utilisateur via la page de confirmation, présentant les types d'informations collectées, ainsi que le détail des informations collectées. Une fois les données soumises, une localkey est générée et stockée dans le localStorage du navigateur, ce afin de reconnaître ce dernier en cas de soumissions multiples. La localkey est aussi affichée à l'utilisateur afin qu'il puisse faire valoir ses droits quant à l'accès et la correction des données le concernant.

4.2 Résultats expérimentaux

A partir des 29 collectes issues de 22 personnes (8 ont été réalisées par la même personne dans des contextes différents), nous allons estimer dans quelle mesure ces informations permettent de mesurer une ressemblance des personnes. La figure 4 présente deux matrices de distance. La première (a) compare les données pré-traitées (sans protection) avec la distance du cosinus ($1 - cos(A, B)$, si A et B sont deux vecteurs de réels). Sur cette figure, nous pouvons

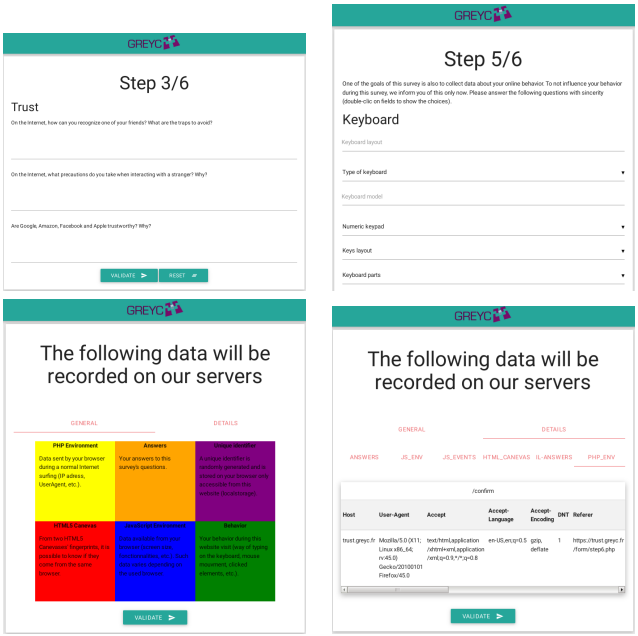


Figure 3 – *Écrans du questionnaire de collecte des données personnelles.*

constater deux choses. La première est que les signatures 4 et 5 sont jugées comme très similaires. Il s'agit en fait de la même personne dans le même contexte. La seule différence est la dynamique de frappe au clavier. Les signatures 3 à 10 ont été générées par le même individu mais dans des contextes différents (wifi, navigateur, ...), la ressemblance est plus contrastée. La seconde constatation importante est la relative similarité des signatures 4 et 5 avec d'autres signatures du tableau. Ceci peut s'expliquer par le fait que les données ont été acquises au sein du laboratoire par du matériel ayant une configuration proche et la même adresse IP sortante.

La figure 4 (b) présente la distance entre les codes binaires (signatures protégées) de la base en prenant pour chaque individu une clé secrète unique. Avec la protection et cette clé, on met en évidence très clairement la similarité entre individus. Pour les codes binaires reliés aux signatures de 3 à 10, on identifie bien une similarité en eux avec des degrés plus ou moins élevés en fonction des données personnelles similaires. Ceci démontre bien la capacité de la méthode proposée à produire un code exploitable pour des calculs de similarité d'informations personnelles.

Concernant les exigences énoncées au début, il est assez facile de vérifier qu'elles sont respectées. La transformation du BioHashing garantie la non inversibilité du code binaire calculé et le respect de la similarité. La confidentialité des données est obtenue par cette dernière transformation et l'usage d'une clé secrète (ici un mot de passe). Un imposteur ne pourra pas générer ce code binaire sans connaître la clé secrète, utiliser le même matériel...Il pourra tout au mieux rejouer une donnée existante. Des mécanismes de protection du canal de communication, et de la donnée côté service, peuvent résoudre ce problème. La révocation du code est aisée en changeant de clé secrète (i.e. ici, mot de passe).

5 Conclusion et perspectives

Dans ce papier, nous proposons une méthode permettant de calculer un code personnel lié à un internaute respectueux de la vie privée. Ce code intègre différentes informations liées à son navigateur, sa façon de taper au clavier, ou sa localisation. Nous avons montré sur une base préliminaire de 29 collectes qu'il était possible d'obtenir un code binaire proche pour la même personne malgré des différences de contexte. Plusieurs applications sont envisageables à ce travail dont l'authentification d'un internaute, l'usage pour identifier des comptes multiples par un service (similarité de codes calculés avec une clé unique). Ces applications constituent les perspectives de cette étude.

Remerciements

Les auteurs tiennent à remercier la région Normandie pour son soutien financier.

Références

[1] Károly Boda, Ádám Földes, Gábor Gulyás, and Sándor Imre. User tracking on the web via cross-browser fingerprinting. *Information Security Technology for Applications*, pages 31–46, 2012.

[2] SL Yinzhi Cao and E Wijmans. Browser fingerprinting via os and hardware level features. *Network & Distributed System Security Symposium, NDSS*, 17, 2017.

[3] Peter Eckersley. How unique is your web browser ? *Privacy Enhancing Technologies*, 6205 :1–18, 2010.

[4] Patrick Lacharme and Aude Plateaux. Pin-based cancelable biometrics. *International Journal of Automated Identification Technology (IJAIT)*, 3(2) :75–79, 2011.

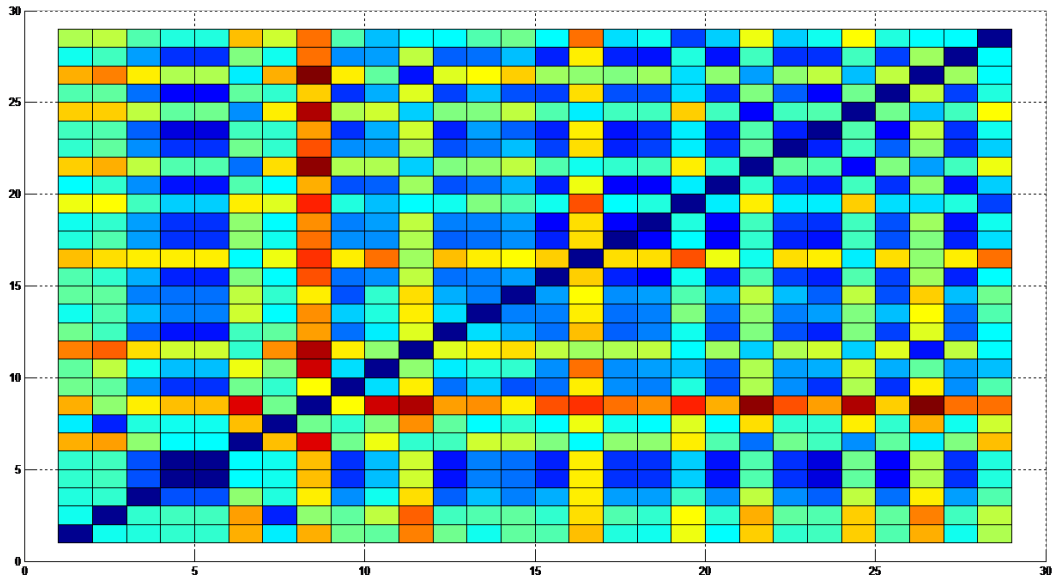
[5] Pierre Laperdrix, Walter Rudametkin, and Benoit Baudry. Beauty and the beast : Diverting modern web browsers to build unique browser fingerprints. *Security and Privacy (SP)*, pages 878–894, 2016.

[6] Nick Nikiforakis, Wouter Joosen, and Benjamin Livshits. Privaricator : Deceiving fingerprinters with little white lies. *Proceedings of the 24th International Conference on World Wide Web*, pages 820–830, 2015.

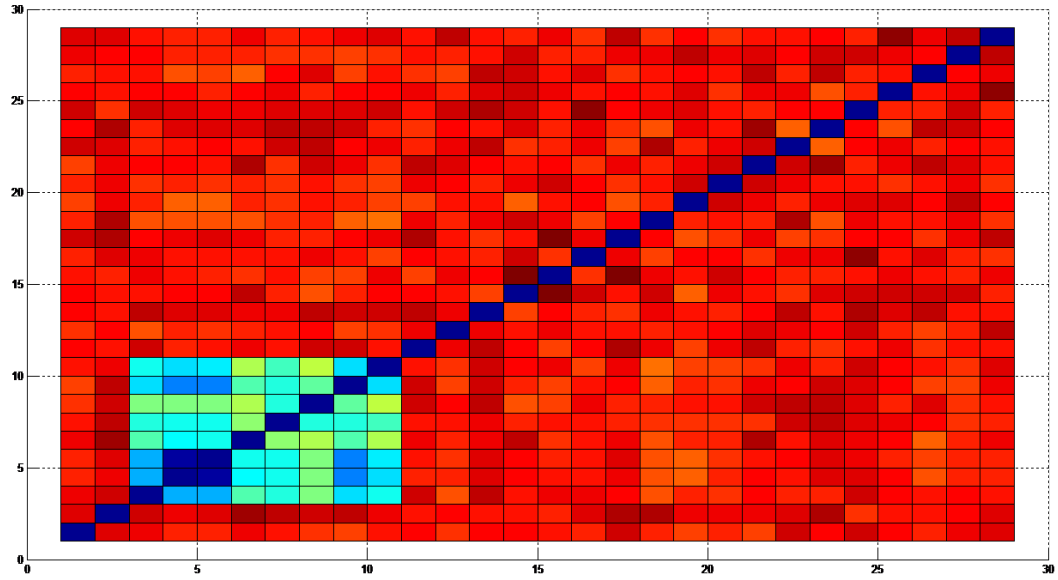
[7] A. B.J. Teoh, Y. W. Kuan, and S. Lee. Cancellable biometrics and annotations on biohash. *Pattern Recognition*, 41 :2034–2044, 2008.

[8] A.B.J. Teoh, D. Ngo, and A. Goh. Biohashing : two factor authentication featuring fingerprint data and tokenised random number. *Pattern recognition*, 40, 2004.

[9] Zachary Weinberg, Eric Y Chen, Pavithra Ramesh Jayaraman, and Collin Jackson. I still know what you visited last summer : Leaking browsing history via user interaction and side channel attacks. *Security and Privacy (SP)*, 2011.



(a)



(b)

Figure 4 – *Représentation de la distance entre les données pré-traitées (a) et après protection (b). En coordonnées, les identifiants des deux entrées comparées (bleu si la similarité est élevée, rouge si faible).*