Decentralizing Privacy: Using Blockchain to Protect Personal Data

G. Zyskind, O. Nathan, A. Pentland
rapport par Yassine Hamoudi

5 janvier 2016

Table des matières

1 Introduction

La collecte et le traitement des données personnelles ont connu une croissance rapide avec le développement d'Internet. De nombreuses entreprises fondent désormais leur activité sur l'exploitation de ces informations (ciblage publicitaire, analyse comportementale, personnalisation des services). Il s'agit alors de garantir simultanément la protection de la vie privée des utilisateurs, et un accès raisonné à leurs données.

La plupart des systèmes actuels sont cependant peu soucieux de ces problématiques. En effet, nos données sont stockées par de nombreux services tiers, sans que nous sachions vraiment où et comment elles sont exploitées. Le risque de les voir compromises et utilisées de manière malhonnête est ainsi important.

Différentes solutions sont développées pour redonner à l'utilisateur le contrôle sur ses informations personnelles. Cela passe d'abord par une démarche éducative, visant à faire prendre conscience des problématiques actuelles en matière de privacité. Sur un plan plus technique, il s'agit de garantir à l'utilisateur une gestion fine des conditions de stockage et d'exploitation de ses données. Comment formaliser la notion de non-divulgation d'informations ? Quels outils développer pour permettre une gestion en temps réel de la diffusion de ses données ?

Nous présentons ici le travail exposé dans [ZNP15]. Les auteurs de cet article proposent une solution décentralisée de stockage des données, s'appuyant sur la chaîne de blocs utilisée par le Bitcoin. Leur protocole garantit à l'utilisateur une gestion fine des droits d'exploitations de ses données (personnalisation selon le service, révocation, ...), combinée aux propriétés de la chaîne de blocs (redondance, protection contre le sabotage, ...).

2 Contexte et état de l'art

La protection des données personnelles connaît un intérêt croissant dans la société civile, et implique de nombreux acteurs (scientifiques, politiques, industriels, ...). Différentes initiatives, rassemblées sous l'acronyme TET (*Transparency Enhancing Technologies*), recensent des solutions concrètes

pour reprendre le contrôle sur ses informations (cf [JVW13]). Un cadre législatif se met également progressivement en place (voir par exemple le travail de la commission européenne sur la protection des données personnelles [1]).

Plusieurs formalisations de l'anonymisation des données ont été proposées, prenant appui sur leur distribution statistique. Par exemple, la $k-$*anonymité* requiert que chaque information soit indistinguable d'au moins $k-1$ autres informations ([Swe02]). D'autres notions existent, comme la $l-$*diversité* ([MKGV07]) ou la $t-$*proximité* ([LLV07]). Malheureusement, ces modèles s'avèrent peu efficaces en pratique. En effet, ce cadre formel s'adapte mal à la variété des attaques possibles. Par exemple, [NS08] présente la désanonymisation d'une base de données Netflix en effectuant un croisement avec les informations publiques contenues dans l'Internet Movie Database (IMDb). Plus généralement, la plupart des informations nous concernant sont si caractéristiques que leur simple divulgation (même anonymisée) permet souvent de remonter à l'individu. [dMHVB13] et [BHF12] détaillent ainsi le profilage d'utilisateurs, basé respectivement sur leurs déplacements et leurs habitudes de navigation internet.

Une solution serait de ne jamais recourir à la divulgation de données brutes, mais autoriser uniquement l'exécution d'algorithmes sur les informations chiffrées (seul le résultat final est accessible aux services). Ceci est formalisé par le chiffrement complètement homomorphe (*Fully Homomorphic Encryption* - FHE), présenté notamment dans [Gen09]. Cependant, les techniques actuelles ne sont pas assez efficaces pour être utilisables en pratique.

3 Protocole de [ZNP15]

Le protocole de [ZNP15] décrit les interactions entre un utilisateur u et un service s (une application mobile par exemple). Ce service souhaite collecter des données sur l'utilisateur. Le protocole a pour objectif d'accorder à u le contrôle sur les informations divulguées.

Un chaîne de bloc est utilisée comme intermédiaire entre ces deux parties. Elle s'occupe de :

— stocker les données communiquées par l'utilisateur dans une table de hachage distribuée (*Distributed Hash Table* - DHT).

— gérer les droits associés (par exemple, l'utilisateur souhaite que sa position ne soit accessible à aucun service)

— redistribuer les données aux services disposant des droits appropriés.

3.1 Description du protocole

Nous disposons d'un schéma de chiffrement symétrique $(\mathcal{G}_{enc}, \mathcal{E}_{enc}, \mathcal{D}_{enc})$ et d'une signature numérique $(\mathcal{G}_{sig}, \mathcal{S}_{sig}, \mathcal{V}_{sig})$. $\mathcal{H}$ désigne une fonction de hachage utilisée par la chaîne de blocs. Cette dernière est représentée par un table de hachage $\mathbb{C}$, de même que la DHT décrite par $\mathbb{D}$. L'ensemble des échanges est signé afin de garantir l'identité de u et s. Nous ne ferons cependant pas figurer dans nos protocoles les vérifications de signatures.

La première étape consiste pour u et s à échanger leurs informations publiques, et partager une clé de chiffrement secrète utilisée pour protéger les données divulguées par u. Cela est effectué de la manière suivante :

Utilisateur		*Service*
$(pk_{sig}^{u,s}, sk_{sig}^{u,s}) \leftarrow \mathcal{G}_{sig}()$		$(pk_{sig}^{s,u}, sk_{sig}^{s,u}) \leftarrow \mathcal{G}_{sig}()$
$sk_{enc}^{u,s} \leftarrow \mathcal{G}_{enc}()$	$\xrightarrow{sk_{enc}^{u,s}, pk_{sig}^{u,s}}$	
	$\xleftarrow{pk_{sig}^{s,u}}$	

Ensuite, l'utilisateur enregistre auprès de la chaîne de bloc le couple qu'il forme avec le service s, ainsi que les autorisations $Droits_{u,s}$ qu'il lui accorde (par exemple, $Droits_{u,s} = \{localisation, contacts\}$) :

1. `http://ec.europa.eu/justice/data-protection/`

Utilisateur		*Chaîne de blocs*
	$\xrightarrow{pk_{sig}^{u,s}, pk_{sig}^{s,u}, Droits_{u,s}}$	$a_{u,s} = \mathcal{H}\left(pk_{sig}^{u,s}\right)$
		$\mathbb{C}[a_{u,s}] \leftarrow pk_{sig}^{u,s}, pk_{sig}^{s,u}, Droits_{u,s}$
		$a_{s,u} = \mathcal{H}\left(pk_{sig}^{s,u}\right)$
		$\mathbb{C}[a_{s,u}] \leftarrow pk_{sig}^{u,s}, pk_{sig}^{s,u}, Droits_{u,s}$

Lorsque l'utilisateur dispose d'une donnée m, il la chiffre via sa clé secrète $sk_{enc}^{u,s}$ et il détermine son type $Type(m)$ (par exemple, $Type(m) = localisation$). Ces informations sont envoyées à la chaîne de bloc qui stock le chiffré c sur la DHT, tout en rendant public un pointeur vers son emplacement :

Utilisateur		*Chaîne de blocs*		*DHT*
$c \leftarrow \mathcal{E}_{enc}(m, sk_{enc}^{u,s})$				
$x_p \leftarrow Type(m)$				
	$\xrightarrow{c, x_p}$	$pk_{sig}^{u,s}, pk_{sig}^{s,u}, Droits_{u,s} \leftarrow \mathbb{C}[\mathcal{H}\left(pk_{sig}^{u,s}\right)]$		
		$a_{x_p} = \mathcal{H}\left(Droits_{u,s} \| x_p\right)$		
		$h_c = \mathcal{H}\left(c\right)$	$\xrightarrow{h_c, c}$	$\mathbb{D}[h_c] = c$
		$\mathbb{C}[a_{x_p}] \leftarrow \mathbb{C}[a_{x_p}] \cup h_c$		
		Diffuser h_c		

Enfin, lorsque le service souhaite accéder à une donnée de type x_p, il communique le pointeur correspondant à la chaîne de blocs. Celle-ci vérifie que s dispose des droits nécessaires, puis lui communique le chiffré c le cas échéant :

Service		*Chaîne de blocs*		*DHT*
	$\xrightarrow{h_c, x_p}$	$pk_{sig}^{u,s}, pk_{sig}^{s,u}, Droits_{u,s} \leftarrow \mathbb{C}[\mathcal{H}\left(pk_{sig}^{s,u}\right)]$		
		Si $x_p \notin Droits_{u,s}$: *échec*		
		$a_{x_p} = \mathcal{H}\left(pk_{sig}^{u,s} \| x_p\right)$		
		Si $h_c \notin \mathbb{C}[a_{x_p}]$: *échec*		
			$\xrightarrow{h_c}$	$c = \mathbb{D}[h_c]$
	$\xleftarrow{c}$		$\xleftarrow{c}$	
$m = \mathcal{D}_{enc}(c, sk_{enc}^{u,s})$				

3.2 Propriétés et sécurité

Le protocole précédant bénéficie des différents avantages de la chaîne de blocs et de la DHT. Le fonctionnement est ainsi décentralisé et redondant. Par ailleurs, tout sabotage nécessite de contrôler la majorité du réseau.

La gestion des droits d'accès se fait via le vecteur $Droits_{u,s}$ communiqué par u à la chaîne de blocs. L'utilisateur peut ainsi modifier les droits à tout moment, et les révoquer en envoyant $Droits_{u,s} = \varnothing$. Cela suppose cependant que s est honnête, et ne conserve pas de données auxquelles il n'aurait plus

accès. Afin de palier à ce problème, il est possible de ne pas accorder d'accès direct aux données, mais de permettre à s d'effectuer des calculs sur celles-ci. Il existe en effet des méthodes distribuées similaires aux FHE, comme la *secure Multiparty Computation* (MPC). Cela peut également être utilisé pour implémenter un vote électronique (fonction de vote distribuée).

Les protocoles de signature protègent contre les usurpations d'identités. Toutefois, si un service est compromis l'utilisateur peut lui retirer tout accès aux données ($Droits_{u,s} = \varnothing$). Enfin, l'utilisation d'une clé $sk_{enc}^{u,s}$ différente pour chaque couple (u, s) limite l'impact d'une attaque, mais implique de chiffrer potentiellement plusieurs fois chaque information.

4 Conclusion

Le protocole décrit dans [ZNP15] permet la mise en place d'un système décentralisé de gestion des données personnelles, basé sur l'utilisation d'un chaîne de blocs. L'utilisateur peut contrôler finement les permissions d'accès à ses données, et gérer la manière avec laquelle elles sont exploitées.

Le système souffre cependant de certaines lacunes qui remettent en cause son utilité pratique. L'utilisation d'une DHT va ralentir les temps d'accès aux données, et son remplacement par du cloud computing par exemple nécessiterait une confiance en un tiers. Par ailleurs, plusieurs étapes du protocole supposent que le service est honnête (par exemple, la révocation des droits n'a pas d'effet si le service a aspirer les données en prévision). La MPC pourrait palier à ce problème, mais on ne dispose pas d'implémentation systématiquement efficaces (réseau important, fonctions complexes) à l'heure actuelle.

Ce protocole demeure tout de même un exemple intéressant d'utilisation de la chaîne de blocs à des fins de privacité.

Références

[BHF12] Christian Banse, Dominik Herrmann, and Hannes Federrath. Tracking users on the internet with behavioral patterns : Evaluation of its practical feasibility. In Dimitris Gritzalis, Steven Furnell, and Marianthi Theoharidou, editors, *Information Security and Privacy Research*, volume 376 of *IFIP Advances in Information and Communication Technology*, pages 235–248. Springer Berlin Heidelberg, 2012.

[dMHVB13] Yves-Alexandre de Montjoye, César A. Hidalgo, Michel Verleysen, and Vincent D. Blondel. Unique in the Crowd : The privacy bounds of human mobility. *Scientific Reports*, 3, March 2013.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pages 169–178, New York, NY, USA, 2009. ACM.

[JVW13] Milena Janic, Thijs Veugen, and Jan Pieter Wijbenga. Transparency enhancing tools (tets) : an overview. In *3rd workshop on Socio-Technical Aspects in Security and Trust (STAST)*, New Orleans, 07/2013 2013.

[LLV07] Ninghui Li, Tiancheng Li, and S. Venkatasubramanian. t-closeness : Privacy beyond k-anonymity and l-diversity. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 106–115, April 2007.

[MKGV07] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. L-diversity : Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1), March 2007.

[NS08] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, SP '08, pages 111–125, Washington, DC, USA, 2008. IEEE Computer Society.

[Swe02] Latanya Sweeney. K-anonymity : A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5) :557–570, October 2002.

[ZNP15] G. Zyskind, O. Nathan, and A. Pentland. Decentralizing privacy : Using blockchain to protect personal data. In *Security and Privacy Workshops (SPW), 2015 IEEE*, pages 180–184, May 2015.