De l'utilisation du chiffrement homomorphe pour calculer dans le domaine chiffré

S. Fau [#††] R. Sirdey [#] C. Fontaine [*††] C. Aguilar-Melchor [†] G. Gogniat [††]

[†] XLIM, Université de Limoges, France
[#] CEA, LIST, France
[††] Lab-STICC, Université de Bretagne Sud, France
[*] CNRS/Telecom Bretagne, France

Résumé

Présentant un surcoût prohibitif à ses débuts, le chiffrement homomorphe offre aujourd'hui, au moins théoriquement, des performances qui permettent de considérer sérieusement son utilisation concrète dans divers contextes.

Cet exposé se veut un premier retour d'expérience sur l'implémentation des systèmes de chiffrement homomorphe et une estimation de son aspect pratique pour effectuer des calculs dans le domaine chiffré.

Mots clés : chiffrement homomorphe, calcul dans le domaine chiffré

I. Introduction

Le chiffrement homomorphe a connu ces dernières des avancées théoriques importantes, dont la plus remarquée est celle de Gentry avec la publication du premier système de chiffrement complètement homomorphe (Fully Homomorphic Encryption) [3]. Durant les années suivantes, beaucoup d'autres systèmes avec différentes propriétés homomorphes ont été présentés, la plupart reposant sur la théorie des réseaux. Parmi ceux-ci, le système de Brakerski et al. [1] (basé sur le problème Learning With Errors) semble être l'un des plus prometteurs en terme de performances théoriques. Malheureusement, très peu d'implémentations de ces systèmes ont été présentées et discutées publiquement, laissant la question de l'utilisation concrète de ces systèmes en suspens.

II. Traitement de données chiffrées

On parle de chiffrement homomorphe lorsque l'on peut effectuer certaines opérations sur les données chiffrées en « passant à travers la couche de chiffrement ». On considère par exemple un système chiffrement probabiliste qui chiffre des bits et dont les chiffrés sont dans un ensemble Ω de grande cardinalité. On souhaite que ce système ait la propriété, pour tout polynôme $p_{\oplus;\otimes} : \mathbb{Z}_2^n \longrightarrow \mathbb{Z}_2$ et son polynôme associé dans le domaine chiffré $p_{\mathsf{add}_\Omega;\mathsf{mul}_\Omega}\Omega^n \longrightarrow \Omega$ (où add_Ω et mul_Ω sont l'addition et la multiplication de chiffrés) :

$$p_{\oplus;\otimes}(m_1, \ldots, m_n) = \mathsf{dec}(p_{\mathsf{add}_\Omega;\mathsf{mul}_\Omega}(\mathsf{enc}(m_1), \ldots, \mathsf{enc}(m_n))).$$

On dira qu'un tel système de chiffrement est complètement homomorphe.

Le système que nous utilisons dans notre implémentation est celui présenté en 2011 par Brakerski et al. [1], qui permet d'évaluer des polynômes à degré limité sur des données chiffrées (polynômes que nous identifions à des algorithmes par la suite). Au-delà de l'implémentation-même de ce système de chiffrement et des performances mesurées, nous souhaitons insister sur les notions de bruit et de profondeur multiplicative (ce qui correspond au degré du polynôme) qui sont primordiales pour la complexité des calculs avec le chiffrement homomorphe.

III. CONTRIBUTIONS

Notre démarche est d'exprimer les algorithmes à haut niveau (grâce aux templates C++) afin de disposer d'un code unique qui permet l'exécution transparente du même algorithme sur des données en clair ou chiffrées [2]. Nous montrons ainsi comment réécrire certaines opérations (<,>,`if-then-else`) afin d'enrichir le langage au-delà de l'addition et de la multiplication effectuées grâce à l'homomorphie du système de chiffrement.

L'exécution de nos algorithmes haut niveau sur des clairs permet de dresser les caractéristiques de ces algorithmes :

	$b^2 - 4ac$ (8 bits)	$\sum_{i=1}^{10} t[i]$ (8 bits)
# add	332	207
# mul	302	135
$\times$ depth	16	8
	b. sort (10×8 bits)	FFT (256×32 bits)
# add	3240	7291592
# mul	2790	5296128
$\times$ depth	136	166

Ces données sont très utiles pour paramétrer le système de chiffrement homomorphe afin qu'il effectue correctement les calculs et avec des performances optimales. Elles permettent aussi de se faire une première idée des performances auxquelles on peut s'attendre lors de l'exécution dans le domaine chiffré.

Nous avons à l'heure actuelle implémenté deux variantes du système de Brakerski et al. (versions vectorielle et polynômiale). Nous donnons quelques résultats expérimentaux obtenus avec de petits paramètres sur des algorithmes évalués sur des chiffrés (dans les deux variantes). Ces résultats permettent de donner une idée de la possibilité d'une utilisation concrète du chiffrement homomorphe, ainsi que des perspectives d'améliorations concernant les performances.

RÉFÉRENCES

[1] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In *ITCS*, pages 309–325, 2012.

[2] S. Fau, R. Sirdey, C. Fontaine, C. Aguilar-Melchor, and G. Gogniat. Towards practical program execution over fully homomorphic encryption schemes. *soumission à WIFS 2012*.

[3] C. Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. crypto.stanford.edu/craig.