Recherche de cliques pour un problème de cybersécurité matériel

Jonathan Fontaine, Lilia Zaourar, Mohamed Benazouz, Roselyne Chotin

Recherche de cliques pour un problème de cybersécurité matériel

Jonathan Fontaine[1], Lilia Zaourar[1], Mohamed Benazouz[1], Roselyne Chotin[2]

[1] CEA Université Paris-Saclay, CEA, List, Bât. 862-PC172, F-91191 Gif-sur-Yvette Cedex, France
{jonathan.fontaine, lilia.zaourar, mohamed.benazouz}@cea.fr, roselyne.chotin@lip6.fr
[2] Sorbonne Université, CNRS, LIP6, F75005, Paris, France

Mots-clés : *Clique, Graphe, Sécurité des circuits intégrés*

1 Introduction

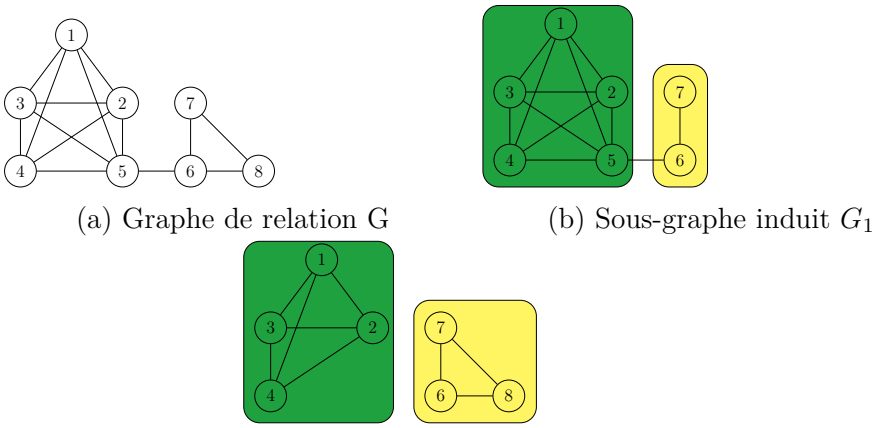
De nos jours, la complexité croissante des circuits intégrés (CI) engendre un coût croissant de fabrication. Afin de contrevenir à cette augmentation, des entreprises se sont spécialisées dans certaines étapes de la chaîne de conception, de la proposition de Propriété Intellectuelle (PI) à la fabrication du CI. Ces entreprises ne peuvent garantir que le CI final correspond exactement aux intentions de l'entreprise conceptrice. En 2018, le magazine Bloomberg businessweek titre «The Big Hack : How China Used a Tiny Chip. to Infiltrate U.S. Companies» nous informe que des chevaux de Troie matériels (CTM) [1] ont été introduits dans des composants destinés aux serveurs d'entreprises américaines. Pour se prémunir des CTMs, il existe différentes méthodes regroupées en deux catégories : prévention et détection. La détection est difficile car les CTMs sont construits pour être furtifs et ne s'activent que dans des conditions précises. Les méthodes de préventions, appelées Design for Hardware Trust modifient le CI, et engendrent généralement un surcoût avec un risque de dégradation des performances.

Nous nous intéressons ici la méthode de *logic locking* qui s'inscrit dans la prévention des CTMs [2]. Son principe est de verrouiller le CI avec une clé numérique, connue uniquement des concepteurs. Chaque bit de cette clé est reliée à une porte logique additionnelle appelée porte clé. Le problème de *logic locking* a pour objectif de maximiser la sécurité tout en minimisant la surface supplémentaire engendrée et l'impact sur le chemin critique.

2 Logic Locking

Le *logic locking* est une méthode générale, et différentes approches ont été développées autour de ce problème. L'une d'elle s'appelle le *Strong Logic Locking* (SLL) [3]. Cette méthode s'intéresse à mesurer la sécurité en fonction de la position des portes clés, en introduisant une notion de relation binaire entre positions, appelé *pairwise secure*. Pour un ensemble de positions, représentant une solution du problème, on peut calculer un graphe de relation. La sécurité est mesurée par : $S(G) = \sum_{c \in C(G)} 2^{|c|}$, avec $C(G)$ l'ensemble des cliques disjointes du graphe de relation. Elle correspond à l'énumération complète de sous-parties inséparables de la clé numérique. De nombreuses heuristiques sont développées afin de résoudre ce problème sans garanties de qualité.

Dans ce contexte, nous proposons ici une formulation mathématique du problème afin de résoudre de façon exacte de petites instances et/ou calculer des bornes. Nous partons de nos travaux précédents [4], et nous proposons de séparer le problème en deux parties, la génération du graphe de relation complet, puis la recherche des positions optimales des portes clés.



(a) Graphe de relation G (b) Sous-graphe induit G_1

(c) Sous-graphe induit G_2

FIG. 1 – Exemple de solution avec K=7

3 Résolution

Nous partons d'une représentation du CI sous forme de graphe de portes logiques appelé *netlist*. Nous construisons en premier lieu le graphe de relation complet $G = (V, E)$, avec V l'ensemble des positions admissibles pour les portes clés, soit les sommets de la *netlist*, et E les relations de *pairwise secure*. Afin de limiter l'augmentation de surface, nous cherchons à limiter le nombre de portes clés à insérer.

Notons K la limite de portes clés, qui est la taille de la clé numérique. Notre objectif est de déterminer la position des K portes clés, en maximisant la sécurité. La sécurité des K positions dépend de la relation de *pairwise*, information synthétisée dans le graphe de relation G. Ainsi, le sous-graphe induit de G par K permet d'obtenir toute l'information nécessaire au calcul de la sécurité de ces K positions. Notre objectif est donc de trouver **le sous-graphe induit** $G' = (V', E')$ tel que $|V'| = K$ et qui maximise la fonction de sécurité S.

Dans la Figure 1, on a un graphe de relation. Posons la limite de la taille du sous-graphe $K = 7$. Deux solutions sont proposées, G_1 avec une clique de taille 5 et une de taille 2, $S(G_1) = 2^5 + 2^2 = 36$, et G_2 avec une clique de taille 4 et une de taille 3, $S(G_2) = 2^4 + 2^3 = 24$.

Deux approches sont évaluées dans ce travail, en partant du graphe de relation complet G et du nombre maximum de portes clés K. La première consiste à énumérer toutes les cliques maximales au sens de l'inclusion de G, puis à sélectionner les cliques intéressantes, avec une approche sac à dos pour les positions choisies. Cette méthode a le défaut d'énumérer toutes les cliques maximales de G. Notre deuxième approche consiste à proposer un PLNE qui cherche un sous-graphe induit, limité par une contrainte de sac à dos, et qui énumère seulement les cliques de ce sous-graphe induit pour mesurer la sécurité. Nous vous présenterons la modélisation et la résolution de ces deux approches.

Références

[1] Xiao, K. and Al. *Hardware Trojans : Lessons Learned after One Decade of Research.* ACM Transactions on Design Automation of Electronic Systems, 2016.

[2] Dupuis, S. and Al. *A novel hardware logic encryption technique for thwarting illegal overproduction and Hardware Trojans.* IEEE IOLTS, 2014.

[3] Rajendran, J. et Al. *Security analysis of logic obfuscation* Design Automation Conference, 2012.

[4] Fontaine, J. et Al. *Optimisation de contre-mesure à l'insertion de Hardware Trojan* ROADEF, 2021.

Ces travaux sont réalisés dans le cadre du projet ANR 18-CE39-0005.