Université de Rennes

Appel à projets - Année 2025

Contrats doctoraux «Ordinaires»

Direction de la Recherche et de l'Innovation

ECOLE DOCTORALE : MATISSE **POLE DE RECHERCHE : Inria/IRISA**

Unité de recherche d'accueil

Unité de Recherche : IRISA

Equipe : PACAP

Encadrement

Co-directeur de thèse : Isabelle Puaut **Unité de recherche** : IRISA

Employeur/Établissement d'origine : Université de Rennes

HDR : ¤ oui • ~~non~~ **Nb de thèses en cours** : 4 (1 à 100%, 3 à 50%)

Mail : isabelle.puaut@irisa.fr **Tél.** : 02 99 84 73 10

Directeur de thèse : Ronan Lashermes **Unité de recherche** : IRISA

Employeur/Établissement d'origine : Inria

HDR : ¤ oui (prévu 1er semestre 2025) • ~~non~~ **Nb de thèses en cours** : 1 à 100 % (se termine en 2025)

Mail : ronan.lashermes@inria.fr **Tél. :**

Co-directeur de thèse : Erven Rohou **Unité de recherche** : IRISA

Employeur/Établissement d'origine : Inria

HDR : ¤ oui • ~~non~~ **Nb de thèses en cours** : 4 (après le 18 février, toutes à 50%)

Mail : erven.rohou@inria.fr **Tél.** : 02 99 84 74 93

Sujet

Titre : **CDIFC: Compilation Durcie pour l'Intégrité du Flot de Contrôle**

Mots clés : Sécurité, Compilation, Microarchitecture

Cotutelle internationale (~~oui~~/non) : **Si oui, préciser l'établissement et le pays de rattachement** :

Descriptif (15-30 lignes) :

Pour se protéger d'un attaquant ayant la capacité de réaliser des attaques par injection de fautes, la microarchitecture des microcontrôleurs doit être conçue avec des contre-mesures spécifiques. Aujourd'hui, cela signifie l'emploi de processeurs en mode *lockstep* : deux cœurs identiques fonctionnent avec les mêmes entrées. En comparant leurs sorties, il devient possible de détecter des fautes simples.

Mais cette solution est insatisfaisante, car elle n'offre pas de garanties fortes de sécurité. C'est pourquoi les chercheurs ont proposé des schémas d'*Instruction Set Randomization* (ISR) [**SOFIA**, **MAFIA**]. Le principe de l'ISR est d'encoder le binaire du programme avec les informations constituant le graphe de flot de contrôle (CFG) autorisé. Le décodage de l'instruction suivante dans le processeur ne devient possible qu'avec un historique d'exécution qui correspond aux chemins possibles dans ce CFG. Des primitives cryptographiques sont utilisées pour garantir la sécurité du schéma.

Malheureusement, aujourd'hui, les schémas d'ISR sont encore trop coûteux (notamment en raison d'un surcoût important en mémoire) et gèrent difficilement certaines instructions. C'est le cas notamment des sauts indirects, c'est-à-dire les instructions de saut qui prennent en argument un registre et donc une donnée. Concrètement, il est généralement impossible de reconstituer un CFG précis efficacement si le programme contient des sauts indirects.

Le sujet de cette thèse sera donc de répondre à la question de recherche suivante : comment supprimer les sauts indirects du code généré par un compilateur ? Quels avantages pouvons-nous obtenir de cette suppression pour la sécurité, et à quel coût au niveau des performances ?

Une première étape de cette thèse sera l'analyse du code binaire généré par un compilateur largement répandu, dans le cadre de cette thèse le compilateur LLVM. Cette première étape répondra aux questions suivantes :

- où sont placés les sauts indirects et pourquoi ?
- Peut-on les remplacer efficacement ?
- Comment réaliser des dispatcheurs efficaces ?
- Peut-on autoriser certains sauts indirects vers l'arrière ?

Dans une deuxième phase, nous modifierons la face arrière du compilateur (*back-end*) pour que le binaire généré ne contienne plus de saut indirect. L'architecture visée sera un processeur RISC-V.

Enfin, la troisième étape consistera à explorer nos nouveaux pouvoirs : l'absence de sauts indirects permettant l'extraction du CFG de manière précise et efficace par le compilateur. Cela autorise des analyses statiques plus précises, dont il restera à évaluer ce qu'elles peuvent permettre pour améliorer la sécurité, notamment en s'inspirant des passes d'analyse statique récemment introduites dans le compilateur GCC.

Les résultats du travail de thèse seront publiés dans les conférences du domaine de la compilation et de la sécurité.

[**SOFIA**] : « Sofia: Software and control flow integrity architecture », R.De Clercq, J. Götzfried, D. Übler, P. Maene, and I. Verbauwhede. Computers & Security, 2017.
[**MAFIA**] : « MAFIA: Protecting the microarchitecture of embedded systems against fault injection attacks », T. Chamelot, D. Couroussé, and K. Heydemann. Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2023.