Projet VÉDYSEC : Vérification Dynamique de Propriétés de Sécurité

Nikolai Kosmatov*, Frédéric Loulergue†, Julien Signoles‡
* Thales Research & Technology, Palaiseau, France
nikolai.kosmatov@thalesgroup.com
† Université d'Orléans, INSA Centre Val de Loire, LIFO EA 4022, France
frederic.loulergue@univ-orleans.fr
‡ Université Paris-Saclay, CEA, List, Palaiseau, France
julien.signoles@cea.fr

Cette soumission présente le projet Astrid Maturation VÉDYSEC (Vérification Dynamique de Propriétés de Sécurité) qui a été soumis et accepté en 2024, et démarre début 2025, avec une durée de 24 mois. Il porte sur les techniques de spécification et vérification à l'exécution de propriétés de sécurité et vise le passage du TRL 4 au TRL 6. Le consortium inclut Thales TRT (coordinateur : Nikolai Kosmatov), le LIFO (responsable scientifique : Frédéric Loulergue) et le CEA List (responsable scientifique : Julien Signoles). Ce projet fait suite aux thèses CEA-DGA de Dara Ly [1] et de Virgile Robles [2].

Contexte

Les progrès récents dans les techniques de vérification des logiciels ont conduit au développement de plusieurs outils de vérification. L'un des plus avancés d'entre eux, Frama-C [3], [4], [5], développé par le CEA List, est un cadre collaboratif *open source* pour l'analyse des programmes C. Il est construit autour d'un noyau qui fournit des services de base aux greffons proposant divers analyses. Les greffons communiquent via des annotations écrites en ACSL (*ANSI C Specification Language*) [6] permettant l'utilisation de plusieurs techniques pour l'analyse et la vérification de programmes C de manière collaborative [7]. L'analyse de valeurs par interprétation abstraite peut être réalisée à l'aide du greffon Eva, la vérification déductive des programmes annotés en ACSL peut être réalisée en utilisant le greffon Wp, la vérification à l'exécution d'annotations ACSL est offerte par le greffon E-ACSL [8].

Cette dernière technique a pour but de traduire une sous-classe des annotations ACSL — celles dites exécutables [9] — en instructions C intégrées au programme sous analyse. Cette transformation permet d'obtenir un nouveau programme C dont la correction vis-à-vis de sa spécification est vérifiée dynamiquement, pendant son exécution. Les annotations sont ainsi vérifiées lors de l'exécution du programme sur des cas de test, ce qui permet de rapporter à l'utilisateur comme verdict tout échec d'annotation détecté (ou l'absence de tels échecs).

La spécification et la vérification d'exigences de haut niveau — telles que des propriétés de sécurité comme l'intégrité ou la confidentialité des données — sur du code de taille conséquente est une activité importante pour l'industrie. Ces propriétés peuvent être exprimées dans Frama-C sous la forme de propriétés globales sur le code (appelées *méta-propriétés*) grâce à une extension d'ACSL, nommée Hilare, permettant d'exprimer des méta-propriétés. Les méta-propriétés peuvent ensuite être traduites en annotations ACSL classiques qui peuvent être vérifiées statiquement ou dynamiquement. Ce cadre conceptuel dans Frama-C a été réalisé sous forme d'un greffon nommé MetAcsl et a été appliqué à plusieurs exemples [10], [11], [12].

Thales a utilisé MetAcsl pour spécifier et prouver les propriétés de sécurité d'une machine virtuelle JavaCard afin de réaliser la certification de la carte à puce de Thales au plus haut niveau d'évaluation (EAL7) des Critères Communs. L'approche MetAcsl s'est avérée à la fois extrêmement pratique, pragmatique et rigoureuse. Le nombre des méta-propriétés étant réduit, il est beaucoup plus simple de les écrire et les relire pour avoir une bonne vision de ce qui est spécifié et vérifié. L'instanciation systématique des méta-propriétés en assertions assure de ne pas passer à côté de la moindre erreur, à condition de prouver ces assertions par la suite. Cette démarche a été reconnue par la communauté scientifique, un CESTI et l'ANSSI [13], [14].

Verrous, objectifs et programme de travail

L'utilisation de MetAcsl avec un objectif de vérification dynamique de proriétés de sécurité a été très peu étudiée. L'utilisation conjointe de MetAcsl avec E-ACSL n'a été que rapidement expérimentée sur des petits exemples [11] et nécessite plusieurs extensions des outils.

Le principal objectif du projet VÉDYSEC est donc le suivant :

$\mathcal{OG}$: la maturation de l'outillage et la consolidation des méthodologies d'application pour la spécification et la vérification dynamique de propriétés de sécurité.

Le programme de travail inclut une étude approfondie des besoins d'extensions, une consolidation de la vérification des annotations dans E-ACSL avec une conception rigoureuse, une extension d'E-ACSL et de MetAcsl pour les propriétés de haut niveau, ainsi qu'une réflexion méthodologique et une évaluation sur des cas d'études ouverts et industriels.

RÉFÉRENCES

[1] D. Ly, "Formalisation d'un vérificateur dynamique de propriétés mémoire pour programmes C," Ph.D. dissertation, Univ. Orléans, 2022. [Online]. Available : https://theses.hal.science/tel-04301178v1

[2] V. Robles, "Specifying and verifying high-level requirements on large programs : Application to security of C programs," Ph.D. dissertation, Univ. Paris-Saclay, 2022. [Online]. Available : https://theses.hal.science/tel-03626084/

[3] F. Kirchner, N. Kosmatov, V. Prevosto, J. Signoles, and B. Yakobowski, "Frama-C : A software analysis perspective," *Formal Asp. Comput.*, 2015.

[4] P. Baudin, F. Bobot, D. Bühler, L. Correnson, F. Kirchner, N. Kosmatov, A. Maroneze, V. Perrelle, V. Prevosto, J. Signoles, and N. Williams, "The dogged pursuit of bug-free C programs : The Frama-C software analysis platform," *Commun. ACM*, 2021.

[5] N. Kosmatov, V. Prevosto, and J. Signoles, Eds., *Guide to Software Verification with Frama-C. Core Components, Usages, and Applications*, ser. Computer Science Foundations and Applied Logic Book Series. Springer, 2024.

[6] P. Baudin, P. Cuoq, J.-C. Filliâtre, C. Marché, B. Monate, Y. Moy, and V. Prevosto, *ACSL : ANSI/ISO C Specification Language*, 2018. [Online]. Available : http://frama-c.com/acsl.html

[7] L. Correnson and J. Signoles, "Combining analyses for C program verification," in *Formal Methods for Industrial Critical Systems (FMCIS)*. Springer, 2012.

[8] J. Signoles, N. Kosmatov, and K. Vorobyov, "E-ACSL, a Runtime Verification Tool for Safety and Security of C Programs," in *Competitions, Usability, Benchmarks, Evaluation, and Standardisation for Runtime Verification Tools (RV-CuBES)*. EasyChair, 2017.

[9] J. Signoles, *E-ACSL : Executable ANSI/ISO C Specification Language*. [Online]. Available : http://frama-c.com/download/e-acsl/e-acsl.pdf

[10] V. Robles, N. Kosmatov, V. Prevosto, L. Rilling, and P. Le Gall, "MetAcsl : Specification and verification of high-level properties," in *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. Springer, 2019.

[11] V. Robles, N. Kosmatov, V. Prevosto, L. Rilling, and P. Le Gall, "Tame your annotations with MetAcsl : Specifying, testing and proving high-level properties," in *Tests and Proofs (TAP)*. Springer, 2019.

[12] V. Robles, N. Kosmatov, V. Prevosto, L. Rilling, and P. Le Gall, "Methodology for specification and verification of high-level properties with MetAcsl," in *Formal Methods in Software Engineering (FormaliSE)*. IEEE, 2021.

[13] A. Djoudi, M. Hána, and N. Kosmatov, "Formal verification of a JavaCard virtual machine with Frama-C," in *Formal Methods (FM)*. Springer, 2021.

[14] A. Djoudi, M. Hána, N. Kosmatov, M. Kříženecký, F. Ohayon, P. Mouy, A. Fontaine, and D. Féliot, "A bottom-up formal verification approach for common criteria certification : Application to JavaCard virtual machine," in *European Congress on Embedded Real-Time Systems (ERTS)*, 2022, best paper award (category "Processes, Methods and Tools"). [Online]. Available : https://hal.univ-lorraine.fr/ERTS2022/hal-03704287v1