Combiner HACL* et BINSEC

analyses de canaux cachés automatiques et incrémentales, et application au cas du post-quantique

Mots clés

méthodes formelles – analyses de canaux cachés – intégration continue – analyses multi-niveau – cryptographie post-quantique

Contact

Aymeric Fromherz, Inria Paris (aymeric.fromherz@inria.fr)
Sebastien Bardin, CEA (sebastien.bardin@cea.fr)
Yannis Sellami, CEA (yannis.sellami@cea.fr)

Contexte

La conception et l'implémentation de bibliothèques cryptographiques de très haute sécurité est un problème central en cybersécurité. Une première étape est de concevoir des primitives et protocoles mathématiquement sécurisés. Une seconde étape, non moins importante en pratique, est de s'assurer que leurs implémentations sont effectivement sécurisées, d'abord d'un point de vue mathématique contre des attaques logiques (aspect fonctionnel : le code implémente correctement les bons concepts cryptographiques), mais aussi contre des attaques de très bas niveau, par exemple des attaques physiques au niveau binaire.

Nous nous concentrons dans ce sujet de recherche sur les attaques par canaux cachés de type timing attacks ou cache attacks. Bien qu'il existe des contre-mesures logicielles à ces attaques (politique de programmation constant-time, dite CT), il est nécessaire de s'assurer que le binaire final respecte bien cette politique, d'une part parce que les développeurs ne la suivent pas toujours (surcoût) – ou peuvent utiliser des composants tiers ne la respectant pas, d'autre part parce que le compilateur peut tout à fait produire un binaire non CT à partir d'un source CT.

Problème

Plusieurs lignes de recherche ont été menées pour apporter des garanties formelles fortes aux implémentations cryptographiques [1, 4, 6], mais malgré des résultats remarqués et des avancées notables, leur utilisation dans un cadre industriel reste limité [5]. Par exemple :

- les techniques d'analyse de code *a posteriori* au niveau binaire [2, 3] permettent de vérifier si un code exécutable respecte CT. Cependant, ces techniques demandent un paramétrage de l'outil (initialisation, définition et identification des secrets) pour chaque code exécutable, ce qui demande une expertise importante et bloque de nombreux utilisateurs. Par ailleurs, ces outils peuvent également souffrir de problèmes de passage à l'échelle car ils doivent raisonner au niveau binaire, ce qui est notablement plus difficile que de raisonner au niveau source [3]
- les techniques de conception et de vérification *by design*, au contraire, cherchent à définir des invariants et propriétés logiques des programmes, souvent exprimées à l'aide de systèmes de types expressifs ou d'annotations, dont la validité implique CT. Cependant, pour utiliser des informations sémantiques fournies par le programmeur, ces analyses opèrent fréquemment sur le code source plutôt que sur le binaire. Il est donc nécessaire de s'assurer que tout compilateur utilisé n'introduise pas de canaux caché, quelle que soit l'architecture ciblée, avant le déploiement et idéalement pendant le processus de développement logiciel.

Objectif de la thèse

Le but général de ce sujet de thèse est de comprendre comment combiner les approches haut niveau de conception formelle de code cryptographique (a priori) avec les approches bas niveau de vérification formelle du code exécutable (a posteriori) afin d'aboutir à la première chaîne de développement formelle de bout en bout permettant d'assurer à la fois la correction des opérations mathématiques implémentées et la résistance à des attaques bas niveau de type canaux cachés, le tout en permettant une expérience utilisateur moderne via le support de mécanismes de CI (automatisation de bout en bout).

Les challenges sont bien sûr de pouvoir obtenir une analyse niveau binaire, précise, efficace, incrémentale et facile à configurer.

Notre idée clé ici est de tirer le meilleur parti des approches et outils BINSEC (analyse de binaire) et HACL⋆ (vérification de code source), pour développer un nouveau cadre d'analyse de programme au niveau binaire *guidé par des annotations et spécifications formelles de haut niveau.* Les questions de recherche comportent notamment l'identification des informations pertinentes – avec potentiellement une extension des primitives présentes dans HACL⋆, leur transmission le long de la chaîne de compilation et leur utilisation effective dans un cadre d'analyse symbolique au niveau binaire.

Nous centrerons notre étude sur le cas des primitives post-quantiques, dont le développement est plus récent, et qui ont donc été bien moins analysées jusque là que leurs homologues classiques d'un point de vue attaques par canaux cachés. Les résultats seront à la fois d'ordre théoriques, méthodologiques, et applicatifs: la méthodologie développée permettra la vérification de bout en bout de primitives cryptographiques, d'une implémentation haut niveau jusqu'à la validation du binaire exécutable; cette méthodologie sera utilisée pour renforcer la fiabilité de la librairie HACL⋆, en particulier à travers le développement d'une CI validant la compilation d'implémentations vérifiées par plusieurs compilateurs grand public et leurs multiples versions, ciblant un grand nombre d'architecture; enfin, l'implémentation inclura des améliorations à l'outil BINSEC, particulièrement pour supporter un plus grand nombre d'architectures, mais également pour interagir plus efficacement avec des binaires existants.

Équipes

BINary-level SECurity research group (BINSEC – `https://binsec.github.io`) membre du CEA List, est un groupe de recherche leader sur l'analyse de code pour la sécurité bas niveau, qui publie régulièrement dans les meilleures conférences internationales en sécurité, en méthodes formelles et en ingénierie logicielle. L'équipe travaille en étroite collaboration avec des équipes de recherches françaises et internationales, des partenaires industriels et des agences nationales.

Inria Prosecco L'équipe Prosecco d'Inria Paris possède une expertise reconnue dans la vérification de protocoles cryptographiques et de leurs implémentations, notamment grâce au codéveloppement de nombreux outils de vérification formelle dont F⋆, CryptoVerif, ProVerif, Squirrel, et Aeneas, mais également d'application cryptographiques vérifiées de haute performance comme la bibliothèque HACL⋆

Encadrement

L'étudiant sera encadré par une équipe conjointe de chercheurs juniors et seniors Inria / CEA. Les deux équipes étant localisées en région parisienne, relativement proches l'une de l'autre, l'étudiant ira régulièrement aux deux endroits.

References

[1] José Bacelar Almeida, Manuel Barbosa, Gilles Barthe, Arthur Blot, Benjamin Grégoire, Vincent Laporte, Tiago Oliveira, Hugo Pacheco, Benedikt Schmidt, and Pierre-Yves Strub. Jasmin: High-assurance and high-speed cryptography. In *CCS*, pages 1807–1823, 2017.

[2] Robin David, Sébastien Bardin, Thanh Dinh Ta, Laurent Mounier, Josselin Feist, Marie-Laure Potet, and Jean-Yves Marion. BINSEC/SE: A dynamic symbolic execution toolkit for binary-level analysis. In *SANER*. IEEE Computer Society, 2016.

[3] Adel Djoudi and Sébastien Bardin. BINSEC: binary code analysis with low-level regions. In *TACAS*. Springer, 2015.

[4] Andres Erbsen, Jade Philipoom, Jason Gross, Robert Sloan, and Adam Chlipala. Simple high-level code for cryptographic arithmetic: With proofs, without compromises. *ACM SIGOPS Operating Systems Review*, 54(1):23–30, 2020.

[5] Jan Jancar, Marcel Fourné, Daniel De Almeida Braga, Mohamed Sabt, Peter Schwabe, Gilles Barthe, Pierre-Alain Fouque, and Yasemin Acar. "they're not that hard to mitigate": What cryptographic library developers think about timing attacks. In *SP*. IEEE, 2022.

[6] Marina Polubelova, Karthikeyan Bhargavan, Jonathan Protzenko, Benjamin Beurdouche, Aymeric Fromherz, Natalia Kulatova, and Santiago Zanella-Béguelin. Haclxn: Verified generic simd crypto (for all your favourite platforms). In *CCS*, pages 899–918, 2020.