Description of the Work (DoW)

PER2023–034 - Type : développement

Programmation de code efficace et sûr de cryptographie

Etudiant(s) : Noah CANDAELE (M2 Informatique Cybersécurité), Anthony IOZZIA (M2 Informatique Cybersécurité)

Encadrant(s) : Sid TOUATI (professeur des universités, INRIA), Benjamin GRÉGOIRE (chargé de recherche INRIA)

1. Résumé exécutif

La cryptographie est un domaine qui nécessite des programmes spécifiques, capables de garantir la sécurité et l'efficacité du code face aux menaces potentielles. Pour répondre à ces besoins, un langage de programmation dédié, nommé Jasmin, a été développé pour créer des programmes cryptographiques bas niveau, performants et sûrs. Libjade, une librairie écrite en Jasmin, se focalise principalement sur les primitives cryptographiques post-quantiques, c'est-à-dire celles qui résistent à un adversaire disposant d'un ordinateur quantique ; elle inclut aussi de la cryptographie classique.

Le langage Jasmin nécessite de connaître l'architecture du processeur sur lequel le code sera exécuté, car il s'agit d'un langage de bas niveau. Ce choix permet de profiter pleinement des performances du matériel en optimisant le code. Cependant, la portabilité du code écrit sera impactée car le code écrit en Jasmin ne sera pas directement exécutable sur d'autres architectures de processeur sans une adaptation significative.

AES-GCM est un algorithme classique très utilisé actuellement, mais il n'a pas encore été implémenté dans Libjade. Le but de ce projet est de réaliser une implémentation d'AES-GCM en Jasmin, et de la comparer avec une ou plusieurs implémentations en C. On s'intéressera d'abord à une version pour l'architecture Intel x86, puis à une version pour x86 avec le jeu d'instructions AES-NI. Enfin, on cherchera à obtenir une version compatible avec un processeur ARMv7.

2. Description du projet

Contexte technologique

- Cryptographie post-quantique : les algorithmes doivent être suffisamment résistants aux futurs ordinateurs quantiques.
- Langage Jasmin : utilisé pour créer des programmes cryptographiques bas niveau et performants.
- Architecture x86 : l'implémentation d'AES-GCM cible d'abord les processeurs Intel x86.
- Jeu d'instructions AES-NI : une phase ultérieure exploitera les instructions spécifiques AES-NI pour optimiser les performances.
- Portabilité limitée : le code en Jasmin nécessitera des adaptations pour être exécutable sur d'autres architectures.
- Langage C : l'implémentation en Jasmin sera comparée à une ou plusieurs implémentations en C pour évaluer les performances.
- Architecture ARMv7 : l'objectif final est de rendre l'implémentation compatible avec les processeurs ARMv7.

Motivations

Ce projet, en utilisant le langage Jasmin, propose une implémentation sécurisée d'AES-GCM. Jasmin permet de certifier le code généré, garantissant ainsi une exécution fiable et robuste de l'algorithme. En comparaison, le langage C, bien que largement répandu, utilise un compilateur complexe qui rend la certification difficile en raison des optimisations avancées pour augmenter ses performances. Ainsi, ce projet offre une alternative plus sûre en offrant une certification plus accessible du code généré, grâce à la simplicité fondamentale du langage Jasmin. Ce projet offre ainsi une démonstration concrète de l'efficacité et de la pertinence du langage Jasmin dans la création de programmes cryptographiques de pointe et contribue significativement à l'avancée de la cryptographie post-quantique. Il s'agit d'une étape cruciale dans la recherche visant à développer des techniques de chiffrement capables de résister aux futurs ordinateurs quantiques, qui représentent une menace potentielle pour les méthodes de chiffrement actuelles.

Objectifs à atteindre

Objectif principal : Développer une implémentation d'AES-GCM pour l'architecture x86 avec le langage Jasmin pour garantir la sécurité, la stabilité et les performances de chiffrement.

Objectifs secondaires :

- Compiler efficacement un code C du même algorithme avec plusieurs compilateurs et options.
- Faire une analyse fine des performances des différentes versions afin de détecter les goulots d'étranglement.
- Modifier éventuellement le code source C pour améliorer ses performances via compilation, et le comparer à la version Jasmin.
- Développer une implémentation d'AES-GCM pour l'architecture x86.
- Utiliser les instructions AES-NI pour améliorer les performances sur l'architecture x86.
- Adapter l'implémentation pour une exécution sur des processeurs ARMv7.

Risques identifiés (et contre-mesures)

- Complexité de l'implémentation en Jasmin : La traduction de l'algorithme AES-GCM en Jasmin pourrait s'avérer complexe et exigeante en termes de temps et de ressources. Pour surmonter cela, une approche méthodique et une compréhension approfondie de l'algorithme seront nécessaires. Il pourrait être utile de découper le processus en étapes plus gérables.
- Certification du code Jasmin : Bien que Jasmin offre la possibilité de certifier le code généré, cela pourrait nécessiter une maîtrise approfondie de l'outil et des mécanismes de certification. Pour faire face à cela, une collaboration avec des experts en sécurité et en certification pourrait s'avérer cruciale.
- Résistance aux ordinateurs quantiques : Même si le projet vise à créer une implémentation résistante aux attaques quantiques, il pourrait exister des vulnérabilités inattendues. Pour atténuer ce risque, des révisions approfondies du code et des tests de sécurité rigoureux seront essentiels.

Scénarios

<u>Scénario d'utilisation 1 : Sécurisation des communications dans une entreprise</u>

Contexte : Une entreprise souhaite utiliser l'implémentation d'AES-GCM en Jasmin pour sécuriser les communications internes et externes.

Critères d'acceptation :

- Compatibilité avec l'architecture x86
- Performances sans latence significative
- Résistance aux (futures) attaques d'ordinateurs quantiques

<u>Scénario d'utilisation 2 : Sécurisation des dispositifs IoT</u>

Contexte : Une entreprise intègre l'implémentation d'AES-GCM en Jasmin dans ses microcontrôleurs ARMv7 pour sécuriser les communications entre les dispositifs IoT et le cloud.

Critères d'acceptation :

- Compatibilité avec l'architecture ARMv7
- Faible utilisation des ressources matérielles pour économiser l'énergie
- Sécurisation des communications IoT vers le cloud

Scénario d'utilisation 3 : Sécurisation des données dans un cloud

Contexte : Un fournisseur de services de cloud computing implémente AES-GCM en Jasmin pour sécuriser les données stockées et en transit.

Critères d'acceptation :

- Compatibilité avec l'architecture des serveurs du fournisseur.
- Performance sans dégradation significative.
- Confidentialité des données clients.

3. Mise en œuvre

Activités déjà accomplies avant les semaines à plein temps :

- Analyse du code existant en Jasmin.
- Expérimentation active du code Jasmin.
- Compréhension approfondie du fonctionnement interne de AES-GCM.
- Clarté sur les objectifs visés pour la librairie Libjade.

Activités planifiées pour chaque semaine à plein temps :

- Implémentation d'AES-GCM en Jasmin.
- Détermination des paramètres optimaux pour la compilation des programmes en C.
- Réalisation de benchmarks entre les programmes en C et notre implémentation en Jasmin.
- Optimisation du code en C et en Jasmin.
- Analyse des différences, avantages et inconvénients de ces deux langages.

Organisation du travail (répartition de l'équipe) :

En raison de la proximité des diverses tâches de ce projet, la méthode de pair programming sera principalement adoptée. Cela garantira la fiabilité du code généré et maintiendra une cohérence logique dans l'avancement des travaux.