

Soutenance de Stage : Les enclaves SGX de Intel

Fonctionnement général, attestation, et applications

Guilhem Niot¹

Encadré par Laurent Réveillère, Simon Da Silva et Pierre Vernaect²

¹Informatique Fondamentale
ENS de Lyon

²Équipe Progress
LaBRI

Juin - Juillet 2020

- 1 Qu'est-ce qu'une enclave ?
- 2 Attestation des enclaves
- 3 Attestation distante + échange de clés
- 4 Applications
- 5 Conclusion

1. Qu'est-ce qu'une enclave ?

À l'origine, un problème de sécurité

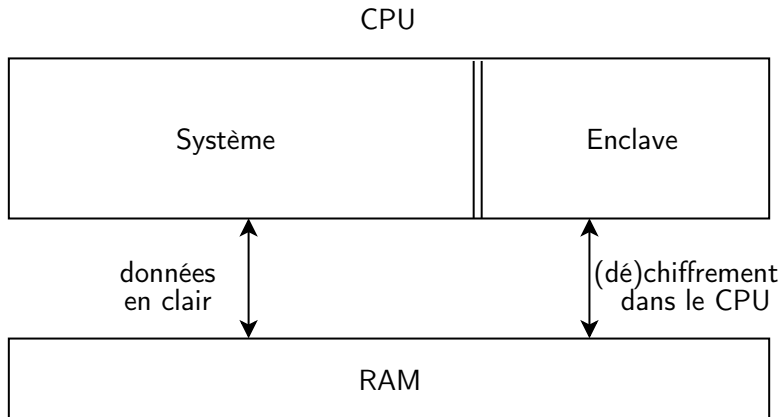
Le système d'enclaves SGX (*Software Guard Extensions*) a été créé par Intel dans le but de permettre l'exécution d'une application sensible dans un environnement géré par un adversaire potentiel.

Par exemple, l'envoi et l'utilisation de clés privées sur un serveur d'un fournisseur tiers tel le cloud AWS de Amazon.

La solution des enclaves

Les enclaves proposent une solution se reposant sur la difficulté de faire des attaques matérielles :

Le flow d'exécution d'une enclave est isolé du flow normal, ses données sont chiffrées dans la mémoire centrale et uniquement déchiffrées dans le CPU lors de leur accès.



Mais notre code s'exécute-t-il bien dans une enclave ?

À priori, rien n'empêche un adversaire d'imiter le jeu d'instructions de Intel et de nous faire croire que notre code s'exécute bien dans une enclave, voire de le modifier avant de le placer dans une enclave.

C'est ici qu'intervient le mécanisme d'attestation.

Chaque CPU Intel possède une clé privée permettant de fournir des preuves de l'authenticité de la plateforme, vérifiables à l'aide des services d'Intel.

2. Attestation des enclaves

Schéma de l'attestation d'une enclave

Pour les raisons évoquées plus haut, à leur initialisation, les enclaves ne contiennent aucune donnée sensible.

Celles-ci sont fournies par un service tiers qui vérifie au préalable que l'application avec laquelle il communique s'exécute bien dans une enclave Intel et que le code exécuté est celui attendu.

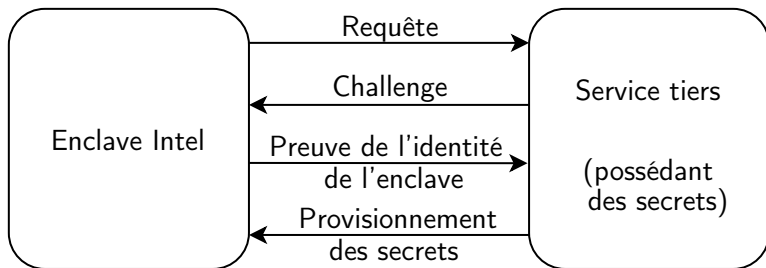


Figure – Schéma d'attestation

Pour vérifier l'exécution d'une application dans une enclave véritable, Intel propose deux niveaux d'attestation :

- L'attestation locale est un mécanisme permettant d'attester une enclave A depuis une autre enclave B s'exécutant sur la même plateforme.
L'identité de l'enclave A est calculée et signée par le CPU avec une clé symétrique accessible uniquement par B.
- L'attestation distante (ou *remote attestation*) permet d'attester une enclave depuis n'importe quelle machine. Elle requiert toutefois de se reposer sur les serveurs de Intel.

Qu'est-ce que l'identité d'une enclave (ou *report*) ?

Dans les deux cas, en plus de l'exécution au sein d'une enclave, on atteste une structure servant d'identité à l'enclave attestée.

Celle ci contient plusieurs champs :

- Le MRENCLAVE qui est un hash du code exécuté dans l'enclave.
- Le MRSIGNER qui est un hash de la clé RSA publique ayant signée le code de l'enclave.
- Le ISV Prod ID (Independent Software Vendors Product ID) est un numéro de produit choisi à la compilation par le fournisseur de l'enclave.
- Le ISV SVN (... Security Version Number) est un numéro correspondant au niveau de sécurité de l'enclave. Il doit être incrémenté à chaque mise à jour de sécurité.
- 64 octets de données utilisateur choisis à l'exécution par l'enclave.

Le processus d'attestation se décompose ainsi en plusieurs étapes :

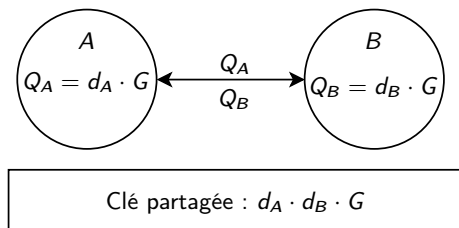
- 1 Demander à l'enclave de fournir des preuves de son exécution sur une plateforme SGX légitime.
- 2 Vérifier ces preuves, soit en local, grâce en à des instructions spécifiques, soit en faisant appel aux serveurs de Intel.
- 3 Accepter ou non l'enclave selon son identité (selon l'auteur, le product ID, et mode non debug par exemple). Cette identité est fournie avec les preuves de l'étape précédente.
- 4 Si l'enclave est acceptée, lui envoyer des secrets.

Protocole de provisionnement de secrets

Puis, pour provisionner l'enclave avec des secrets, il faut établir un canal sécurisé entre l'enclave attestée et le service attestant.

Le SDK de SGX fournit un protocole Diffie-Hellman adapté aux attestations locales et distantes pour établir ce canal.

Chaque parti génère une paire publique/privée de clés elliptiques. Ils échangent leur clé publique, et cela leur permet de calculer une clé privée commune.



G : générateur du groupe elliptique
 d_X : clé privée de X
 Q_X : clé publique de X

Calcul de la clé partagée

Pour calculer $d_A \cdot d_B \cdot G$,

- l'enclave A calcule $d_A \cdot Q_B = d_A \cdot (d_B \cdot G)$
- l'enclave B calcule $d_B \cdot Q_A = d_B \cdot (d_A \cdot G)$

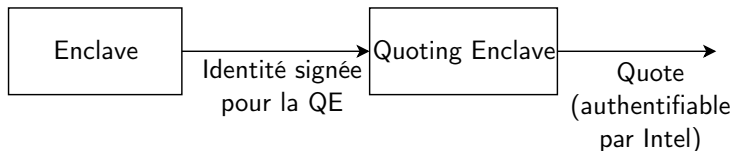
3. Attestation distante + échange de clés

Attestation distante

Lorsque l'on souhaite provisionner une machine dans laquelle aucune enclave de confiance ne s'exécute déjà, on fait une attestation distante.

On se repose dans ce cas sur un tiers de confiance : Intel. L'enclave à attester s'identifie localement auprès d'une enclave de Intel : la Quoting Enclave (ou QE). Pour cela, une attestation locale est réalisée.

Cette Quoting Enclave signe ensuite l'identité de notre enclave avec une clé dont l'accès est restreint localement, et authentifiable à l'aide des serveurs de Intel.



Intel fournit deux modes d'attestation distante :

- Le mode historique, publié en 2016 en même temps que SGX, est basé sur des clés EPID (*Enhanced Privacy ID*). Ce mode requiert un appel systématique (à chaque attestation) aux serveurs d'Intel.
- Le mode ECDSA (*Elliptic Curve Digital Signature Algorithm*), introduit par SGX DCAP (*DataCenter Attestation Primitives*), permet de limiter la dépendances à Intel à une récupération occasionnelle d'un certificat par machine.

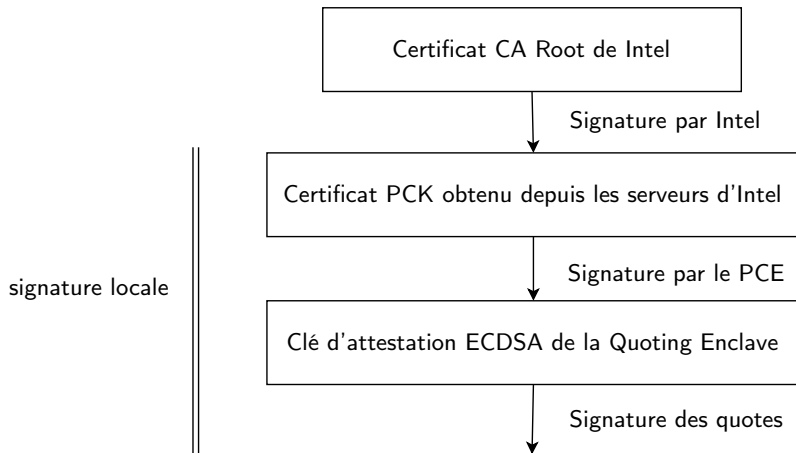
Pour l'attestation ECDSA, la Quoting Enclave génère une clé ECDSA et la fait signer par une autre enclave : l'enclave PCE (*Provisioning Certification Enclave*).

L'enclave PCE utilise pour cela une clé PCK (*Provisioning Certification Key*) qui est dérivée de la Root Provisioning Key.

Cette clé PCK est ainsi calculable par Intel. Intel expose alors un certificat permettant de vérifier ses signatures depuis n'importe quel serveur.

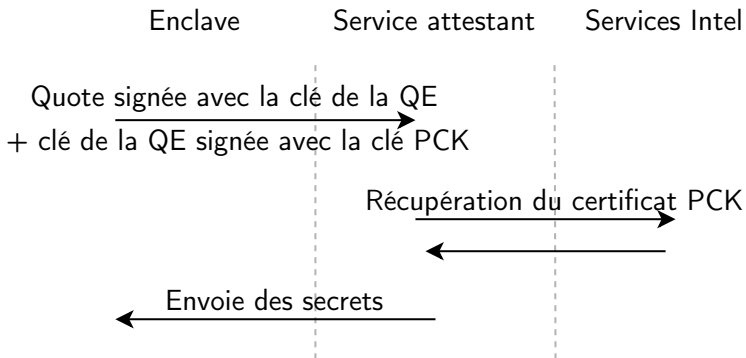
Les Quoting Enclave : Attestation ECDSA

On obtient une hiérarchie de clés :



Les Quoting Enclave : Attestation ECDSA

On peut résumer l'attestation ECDSA ainsi :



Limitation des requêtes à Intel

Pour limiter les requêtes à Intel dans ce schéma d'attestation, l'idée est de mettre en cache les certificats PCK.

Ces certificats sont en effet liés aux niveaux de sécurité matériels et logiciels des machines attestées et ont une durée de vie longue.

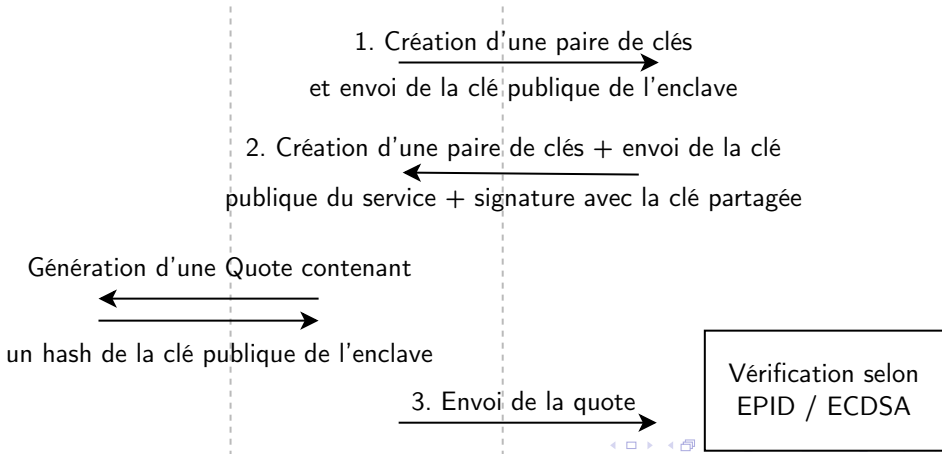
Protocole d'attestation distante + échange de clés

Le protocole ressemble à celui local. Toutefois, l'identité du serveur distant est vérifiée par une clé codée en dur dans l'enclave.

Quoting Enclave

Enclave

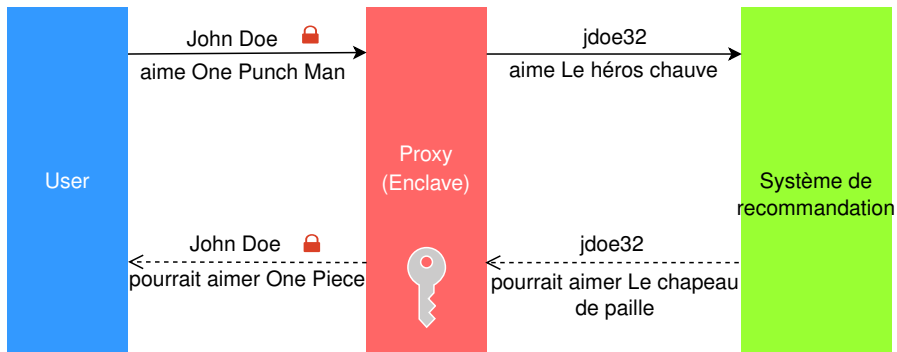
Service attestant




4. Applications

PProx : un système de recommandations anonymisé

L'équipe Progress a conceptualisé un système de proxys (s'exécutant dans des enclaves) transformant les requêtes de l'utilisateur en pseudonymes :



 Communication chiffrée

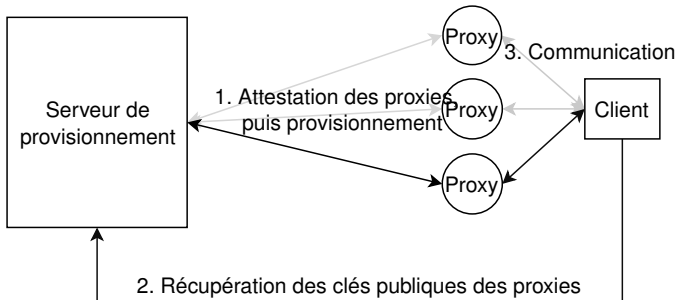
 Clé pour les communications chiffrées +
clé symétrique pour les pseudonymes

Montée en charge de PProx

Le but principal du stage était de créer une infrastructure permettant la montée en charge de PProx.

Il faut pour cela pouvoir instancier de nouvelles enclaves à la volée et les provisionner avec les clés de communication et d'anonymisation.

Pour que ces clés restent protégées, nous avons opté pour l'utilisation d'une enclave de provisionnement qui atteste les proxys avant de les provisionner :



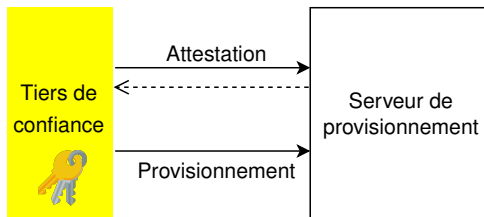
Authentifier le serveur de provisionnement depuis le client

Le client devant récupérer les clés de communication/d'anonymisation auprès du serveur de provisionnement, il doit s'assurer de son identité. Une solution naturelle serait de faire une attestation SGX.

Toutefois, les attestations ont une durée non négligeable (environ 500ms pour l'attestation ECDSA et 2s pour EPID), ce qui ne nous a pas semblé acceptable pour une opération aussi courante qu'est la connexion d'un utilisateur.

Authentifier le serveur de provisionnement depuis le client

L'idée a ainsi été de faire intervenir un tiers de confiance (un serveur possédé en propre ou un ordinateur de l'organisme par exemple) qui transmet une clé privée au serveur de provisionnement, permettant ensuite son identification auprès des clients.



Clé asymétrique identifiant le serveur de provisionnement

5. Conclusion

Ce stage aura permis de synthétiser des documentations éparses pour l'équipe Progress et de mettre en place une infrastructure fonctionnelle permettant une scalabilité efficace pour PProx, ainsi que pour des applications semblables s'exécutant dans des enclaves.

Plusieurs points restent à améliorer :

- Le serveur de provisionnement constitue un single point of failure.
- Il est possible de diminuer l'exposition de la clé d'identification du serveur de provisionnement (qui a une durée de vie longue) avec un mécanisme de dérivation tel BIP 32¹.

Les explications et applications établies lors de ce stage seront réutilisées et approfondies lors de la thèse CIFRE de Pierre Vernaect, doctorant de Laurent Réveillère.

1. <https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki>