-- Attestation distante d'intégrité sous Android -- Dimitri Kirchner -- AMOSSYS --

Abstract

En informatique de confiance, le mécanisme d'attestation distante permet à un logiciel de prouver son intégrité à une entité tierce. Cette entité est alors particulièrement sensible car elle contient les données de référence qui permettent de valider ou d'inférer l'intégrité du logiciel mesuré. Le principe de ses travaux est donc d'offrir à l'utilisateur la possibilité de baser sa confiance sur une entité qui ne quitte que rarement la poche de son propriétaire : son *smartphone*. Contrairement aux solutions classiques d'attestation d'intégrité, l'architecture proposée est dépendante d'aucune infrastructure de confiance sur le réseau et repose sur le simple branchement par USB de son téléphone à un poste utilisateur. Pour cela, nous avons utilisé un téléphone Android afin d'attester de l'intégrité d'une plateforme Debian disposant d'une puce TPM[1]. La preuve de concept développée consiste à se baser sur le mécanisme de *tethering* USB proposé par Android, ainsi que sur la solution de sources libres OpenPTS[2].

Mots clés : Trusted Computing, Remote Attestation, Integrity Verifier, Android, OpenPTS.

Rappels sur l'informatique de confiance et l'attestation d'intégrité

L'objectif de l'informatique de confiance est de permettre d'attester de manière locale ou distante de l'intégrité d'un logiciel, et plus généralement d'une plateforme. Pour cela, l'architecture de sécurité définie par le *Trusted Computing Group*[3] repose sur le composant cryptographique TPM. Celui-ci met à disposition de l'utilisateur des fonctions cryptographiques, de stockage sécurisé, mais aussi d'agrégation de mesures, mécanisme qui consiste à concaténer la valeur courante d'un registre avec la mesure réalisée par une application logicielle, puis à hacher cryptographiquement le tout (en utilisant la fonction de hachage SHA-1). Ce mécanisme utilise des mémoires internes au TPM et uniquement accessibles par celui-ci : les registres PCR.

Les registres PCR permettent de caractériser l'intégrité d'une plateforme, en agrégeant les mesures des composants du système jugés sensibles, par exemple des éléments de la chaine de démarrage, des pilotes noyau, des fichiers de configuration, etc. Ces mesures sont réalisées par des services de confiance, qui agrègent les hachés cryptographiques de ces informations sensibles dans les registres PCR du TPM (**figure *1***). Dès lors que l'un des éléments mesurés est modifié de manière légitime ou non, la valeur des registres PCR sera alors impactée.

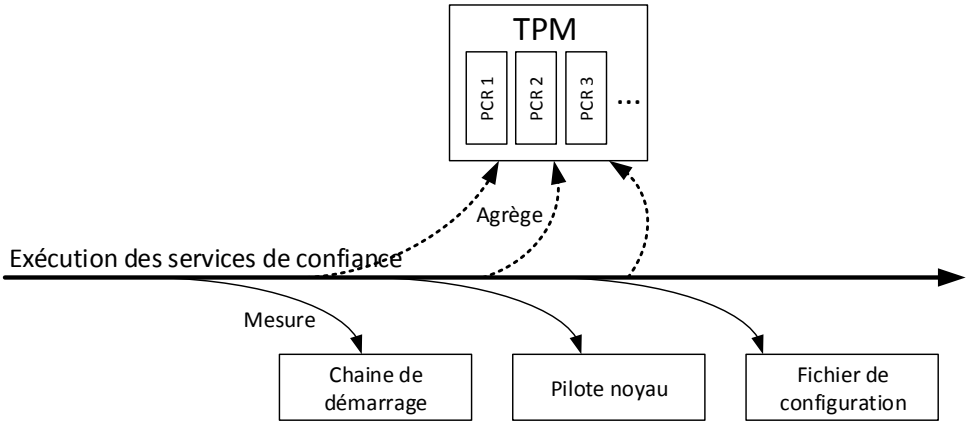

Figure 1 : Mesure et agrégation des informations sensibles de la plateforme dans les registres PCR du TPM.

Le protocole d'attestation distante permet à une plateforme (appelée *Collector*) de prouver son intégrité à une entité tierce distante (appelée *Verifier*). Le protocole se décompose en deux phases distinctes. La première, appelée phase d'initialisation, permet au *Verifier* d'enregistrer la clé publique du TPM associée à la plateforme à attester, ainsi que les valeurs des PCR caractérisant un état intègre.

Une fois cette initialisation réalisée, la phase de vérification permet d'attester de l'intégrité et de l'identité du *Collector*. Pour cela, le *Verifier* effectue une demande d'attestation, afin de récupérer les valeurs actuelles des registres PCR. Afin d'attester de l'identité de la plateforme, et afin de protéger en intégrité les valeurs extraites des registres PCR, le protocole d'attestation demande à la puce TPM de signer numériquement ces valeurs. Le *Verifier* peut ensuite valider l'authenticité du TPM, puis comparer l'état des PCR extraits par rapport aux valeurs de référence ayant été enregistrées lors de la phase d'initialisation (**figure *2***). Si celles-ci diffèrent, le poste client n'est alors pas reconnu comme intègre.

---

[1] *Trusted Platform Module* : composant cryptographique matériel sur lequel repose l'informatique de confiance.
[2] *Open Platform Trust Services* : http://openpts.sourceforge.jp/
[3] http://www.trustedcomputinggroup.org/

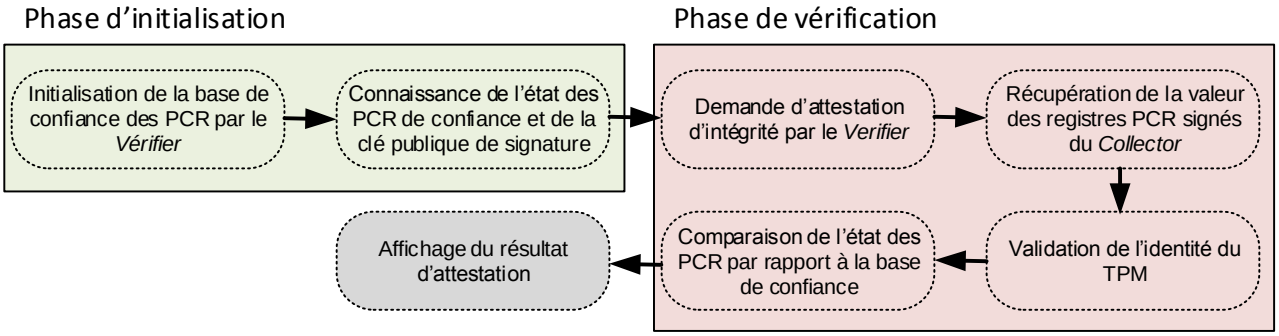

Figure 2 : Cinématique des étapes d'initialisation et de vérification du protocole d'attestation distante.

Les différentes implémentations d'attestation à distance proposées aujourd'hui reposent sur la vérification de l'authenticité du TPM par une infrastructure à clés publiques. Il est alors nécessaire de disposer d'une telle infrastructure accessible sur le réseau dans le but d'attester de l'intégrité ne serait-ce que d'un poste. Afin que l'architecture de la solution proposée soit adaptée à la vérification d'intégrité d'un poste autonome, ces travaux reposent donc sur un simple branchement par USB de son téléphone au poste utilisateur.

Architecture de la solution d'attestation distante sous Android

La réalisation de ces travaux est issue de la question suivante : comment s'assurer qu'un poste utilisateur ayant été laissé pendant une durée indéterminée sans protection physique (*i.e.* vulnérable à une attaque de type *Evil Maid attack*) ou logiciel, dispose toujours d'un système intègre ? La solution proposée est de connecter par USB son téléphone Android au poste utilisateur, et de demander au *smartphone* d'attester ou d'inférer de l'intégrité du poste.

La solution de sources libres OpenPTS[2] (pour *Open Platform Trust Services*) est une preuve de concept implémentant le protocole d'attestation distante décrit précédemment, et reconnue comme implémentation de référence par le TCG[4]. Celle-ci se compose de deux principales briques logicielles :

- Le composant nommé *ptsc* implémente la logique relative au *Collector*. Pour cela, la pile logicielle du système d'exploitation communiquant avec le TPM doit avoir été préalablement installée et configurée.
- Le composant nommé *jopenpts* implémente toute la logique relative au *Verifier*. Celle-ci étant développée dans le langage Java, seule la présence d'une JVM (*Java Virtual Machine*) est nécessaire.

Pour communiquer entre ces deux composants, la solution OpenPTS repose sur le protocole SSH. L'installation d'un serveur OpenSSH sur le *Collector* permet alors de disposer de toute la logique nécessaire à l'attestation distante du poste utilisateur par le *Verifier*.

A*ndroid-attest* est une application Android développée dans le cadre de ces travaux, qui effectue le rôle de *Verifier* au sein de l'architecture précédemment détaillée. L'utilisation de la machine virtuelle Dalvik permet de se reposer sur une version adaptée de la logique de *jopenpts*, où les données de référence sont enregistrées dans la mémoire interne du téléphone. D'autre part, afin que l'application puisse communiquer avec le *Collector*, celle-ci repose sur le mécanisme de *tethering* USB permettant de partager la connexion réseau d'un terminal Android avec un périphérique connecté en USB. Une interface réseau dédiée est ainsi initialisée, et une adresse IP associée au poste utilisateur permet d'adresser celui-ci. La connexion SSH au *Collector* est alors réalisée à l'aide de la librairie Java *Jsch* embarquée dans l'application.
La **figure *3*** détaille l'intégration de l'application *android-attest* au sein de l'architecture d'attestation distante.

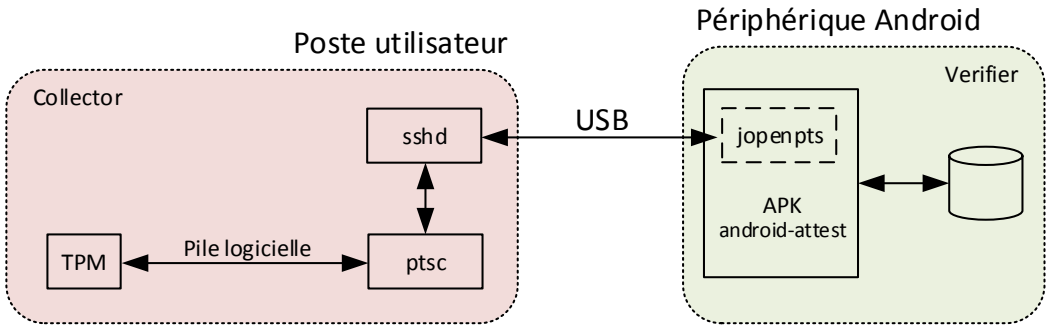

Figure 3 : Architecture de la solution d'attestation distante sous Android.

[4] http://www.trustedcomputinggroup.org/resources/open_platform_trust_service

Finalement, l'application Android développée lors de ces travaux permet d'attester de l'intégrité d'un poste utilisateur autonome, en connectant simplement par USB son *smartphone* à celui-ci. Aucun mécanisme d'attestation distante de ce type n'étant proposé à l'heure actuelle à la connaissance de l'auteur, cet outil est destiné à être publié sous la forme d'une preuve de concept, et par exemple disponible à travers un dépôt GitHub.