Analyse de sécurité des applications de traçage pour COVID-19

Maria ZHEKOVA
University of Luxembourg
maria.zhekova.001@student.uni.lu

Marjan SKROBOT
University of Luxembourg
marjan.skrobot@uni.lu

– **BSP6** –
Résumé
June 26, 2021

1| Introduction

De nos jours, la pandémie de COVID-19 est devenue l'un des sujets de discussion les plus courants. Pour aider à lever en toute sécurité les restrictions de santé publique actuelles, de nouvelles technologies appelées applications de traçage de corona sont en cours de développement. Avec le développement de ces applications et la façon dont les utilisateurs sont suivis, des risques de sécurité et de confidentialité apparaissent. D'où la raison de notre projet.

2| Description du projet

Dans cet article, nous présentons le résumé du projet de semestriel qui avait pour objectifs d'étudier différentes architectures d'application de traçage corona et de concevoir un modèle qui peut être utilisé dans un vérificateur de modèles pour une analyse et une vérification plus approfondies.
Nous utiliserons un vérificateur de modèles appelé Uppaal [1].

3| Résultats

Dans cette section, nous allons brièvement présenter les résultats du projet.
Un système de traçage consiste principalement des utilisateurs, autorité sanitaire et un serveur.
Nous n'avons pas conçu le serveur, ni l'autorité. Par conséquent, notre système consiste uniquement à ce que les utilisateurs échangent des identifiants éphémères `EphID` en tant que messages de rencontre. Le système a été vérifié à l'aide de deux requêtes.

3.1 Utilisateur

Au début l'utilisateur choisit un lieu et un statut COVID-19 qui peut être positif, négatif ou éventuellement positif. L'ensemble des `EphID` est prédéfini et l'utilisateur doit choisir en début de journée son `EphID` en vérifiant du nombre maximal possible de `EphID`s par jour. Si possible, l'utilisateur peut changer son `EphID` ou passer au jour suivant en répétant le processus depuis le début.

3.2 Adversaire

Dans notre système, nous avons conçu deux adversaires.
Un des adversaires peut capturer les rencontres diffusées de partout et les transmettre à tous les emplacements possibles, sans tenir compte de l'emplacement de l'adversaire.
L'autre adversaire ne peut capturer les rencontres qu'à partir du même emplacement que l'adversaire lui-même, mais lorsqu'il change d'emplacement, l'adversaire peut diffuser les rencontres déjà capturées d'un autre emplacement vers le nouveau.

3.3 Requête

Afin de vérifier le système conçu, nous avons utilisé deux requêtes pour deux vérifications similaires.
Le premier vérifie si chaque fois qu'un utilisateur reçoit un `EphID`, il est reçu d'un autre utilisateur et non de lui-même et s'il s'agit d'une communication bidirectionnelle.
La deuxième requête est similaire à celle auparavant, mais elle recherche également aussi l'emplacement. Ainsi, il vérifie à nouveau la même propriété que ci-dessus mais aussi si l'`EphID` reçu provient d'un utilisateur au même endroit.

Le modèle de vérification avec plus de détails en anglais peut être trouvé sur notre dépôt GitHub [2]

4| Conclusion

Les objectifs ont été remplis et nous avons pu livrer un système contenant un modèle des utilisateurs et de deux adversaires.
En suivant la méthode de vérification du système, nous avons pu vérifier avec succès notre système pour les propriétés données à l'aide des deux requêtes implémentés.
Par conséquent, nous pouvons conclure le projet comme un succès.

References

[1] Department of Information Technology at Uppsala University, Sweden & the Department of Computer Science at Aalborg University in Denmark, "About Uppaal", [En-ligne]. Disponible: `https://uppaal.org` [Accédé le 20 juin 2021]

[2] M.Zhekova, M.Skrobot, Y.Kim, "Security analysis of Corona Tracing Applications", [En-ligne]. Disponible: `https://github.com/mzhekova97/BSP6-zhekova-maria` [Accédé le 26 juin 2021]