J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr

Hyperledger Composer 기반 컨소시움 블록체인을 이용한 위조 모바일 APK 검출 DApp☆

Consortium Blockchain based Forgery Android APK Discrimination DApp using Hyperledger Composer

이 형 우1* 이 한 성1
Hyung-Woo Lee Hanseong Lee

요 약

APK (Android Application Package)는 리패키징 공격에 취약하므로 APK 파일 내부에 난독화 기술이 적용되어 있다. 하지만 리버스 엔지니어링 기술 역시 더욱 고도화 됨에 따라 또다른 위조 모바일 APK 파일이 개발 및 배포되고 있어 새로운 대응 방식이 필요하다. 블록체인은 암호화 방식을 사용하여 연결 및 보호되는 레코드 블록이 지속적으로 추가되는 방식으로, 각 블록에는 일반적으로 이전 블록의 암호화 해시값, 타임스탬프 및 트랜잭션 데이터 등을 포함하고 있다. 따라서, 일단 블록체인에 기록되면 해당 블록의 데이터는 이후에 생성된 모든 블록을 변경하지 않고서는 소급해서 변경/수정할 수 없다. 그러므로 블록체인 기술을 적용하면 모바일 APK 파일에 대한 정상 및 위조 여부를 확인할 수 있다. 이에 본 논문에서는 Hyperledger Composer를 이용한 컨소시움 블록체인 프레임워크를 기반으로 합법적인 APK를 블록체인 내에 기록하고 유지함으로써 위조 APK에 대한 검출 기능을 제공하는 DApp (분산형 애플리케이션)을 개발하였다. 제안된 DApp을 통해 사용자의 스마트폰에 위조된 앱이 설치 되는 것을 사전에 방지 할 수 있으므로 궁극적으로는 정상적이고 합법적인 안드로이드 모바일 앱 사용 환경을 제공할 것으로 기대된다.

☞ 주제어 : 안드로이드 모바일 APK, 검출 DApp, 위조 판별, 블록체인, 하이퍼레저 컴포저.

ABSTRACT

Android Application Package (APK) is vulnerable to repackaging attacks. Therefore, obfuscation technology was applied inside the Android APK file to cope with repackaging attack. However, as more advanced reverse engineering techniques continue to be developed, fake Android APK files to be released. A new approach is needed to solve this problem. A blockchain is a continuously growing list of records, called blocks, which are linked and secured using cryptography. Each block typically contains a cryptographic hash of theprevious block, a timestamp and transaction data. Once recorded, the data inany given block cannot be altered retroactively without the alteration of all subsequent blocks. Therefore, it is possible to check whether or not theAndroid Mobile APK is forged by applying the blockchain technology. In this paper, we construct a discrimination DApp (Decentralized Application) against forgery Android Mobile APK by recording and maintaining the legitimate APK in the consortium blockchain framework like Hyperledger Fabric by Composer. With proposed DApp, we can prevent the forgery and modification of the appfrom being installed on the user's Smartphone, and normal and legitimate apps will be widely used.

☞ keyword : Android Mobile APK, Detection DApp, Forgery Detection, Blockchain, Hyperledger Composer.

1. Introduction

A blockchain[1,2] is a continuously growing list of blocks linked and secured using cryptography. It is "an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way". For use as a distributed ledger, a blockchain is typically managed by a peer-to-peer network collectively adhering to a protocol for validating new blocks. Once recorded, the data in any given block cannot be altered retroactively without the alteration of

1 Div. of Computer Engineering, Hanshin University, Gyeong-gi (Osan),18101, Rep. of Korea.
* Corresponding author (hwlee@hs.ac.kr)
[Received 21 June 2019, Reviewed 06 August 2019, Accepted 23 August 2019]
☆ A preliminary version of this paper was presented at ICONI 2018. And this work was supported by Hanshin University Research Grant. This work was partially supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) (NRF-2017R1D1B03035040).

all subsequent blocks, which requires collusion of the network majority. Therefore, it is possible to check whether or not the Android-based mobile App is falsified by Applying the blockchain technology.

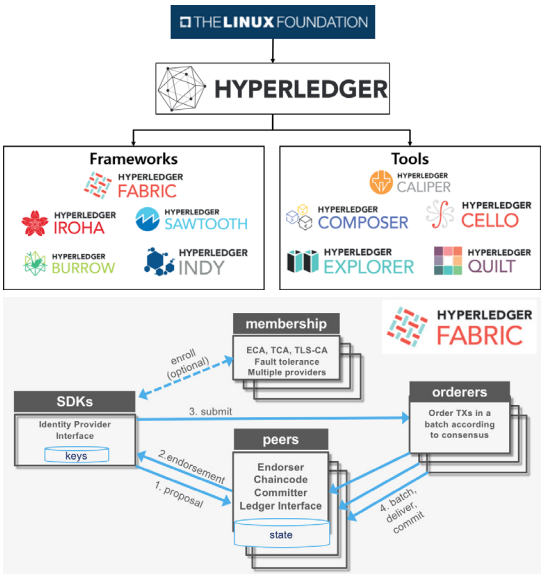
To address the problem of detecting malicious Apps and extracting the corresponding evidence in Android mobile devices, it is possible to use a consortium blockchain framework, which is composed of a detecting consortium chain shared by peer members and a public chain shared by users. In first, we perform feature extracting and modeling by utilizing statistical analysis method, so as to extract legal APK features, including software package feature, permission and application feature, and internal Software Development Kit (SDK) feature. And then, we implemented a discrimination and verification mechanism against mobile fake Android mobile app using consortium blockchain technology such as Hyperledger[3] framework. In detail, we implemented a blockchain based forgery android mobile APK detection DApp(Decentralized Application) by using the Hyperledger Composer[4] on private consortium blockchain Hyperledger Fabric[5] platform and it was possible to determine whether it is forged and counterfeit Android mobile apps correctly.

In Section 2, we present a blockchain structure based on Hyperledger. In Section 3, we proposed a smart contract based Android fake APK detection method. In Section 4 and 5, conclusions are presented together with the design and implementation results of consortium blockchain based forgery Android APK discrimination DApp using Hyperledger Composer for counterfeited APK detection.

2. Hyperledger for Blockchain

2.1 Hyperledger Greenhouse

Hyperledger is an open source project developed by the Linux Foundation(LF) to develop enterprise blockchain technology. Based on the Hyperledger umbrella strategy, Hyperledger introduced a strategy to reinforce the community by reusing common infrastructure elements and to rapidly develop Distributed Director Technology (DLT).Each framework and tool is based on the project life cycle and is determined bythe Technical Steering Committee(TSC). Currently, Hyperledger consists of frameworks such as Fabric, Indy, and Tools such as Composer and Explorer as shown in Figure 1.



(Figure 1) Hyperledger Frameworks and Tools[3]

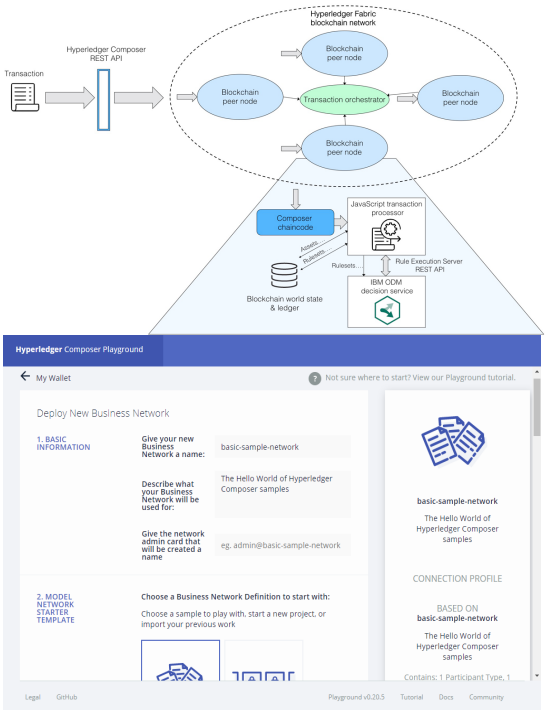
Hyperledger Fabric provides DLT and Smart Contract[6,7] Engine. Core technologies such as consensus and membership services are implemented in a plug-and-play manner based on Java / Go / Node.js language. To implement a mechanism to judge whether a normal application is registered and to compare APK, authorized organizations only have the ability to register and verify normal apps. As a processing speed should be fast, proposed DApp was developed based on Linux Foundation's Hyperledger Fabric, which is one of Consortium blockchain method.

Hyperledger Composer is a tool used to model and build a blockchain network. It provides various functions to easily develop various types of blockchain applications based on the Hyperledger Fabric framework. In addition, Hyperledger Explorer[8] provides a search function to check blocks, transactions, related data, network information, and generated chain code information in the form of a web application. Therefore, in this paper, we apply the Hyperledger Fabric framework using the Hyperledger Composer to check whether the Android mobile APK is forged or not, and verify the chain code information developed and generated through the

Hyperledger Explorer.

2.2 Hyperledger Composer

Hyperledger Composer is an extensive, open development toolset and framework to make developing blockchain applications easier. Our primary goal is to accelerate time to value, and make it easier to integrate our blockchain applications with the existing business systems. We can use Composer to rapidly develop use cases and deploy a blockchain solution in weeks rather than months. Composer allows us to model our business network and integrate existing systems and data with your blockchain applications as shown in Figure 2. Hyperledger Composer supports the existing Hyperledger Fabric blockchain infrastructure and runtime, which supports pluggable blockchain consensus protocols to ensure that transactions are validated according to policy by the designated business network participants [8].



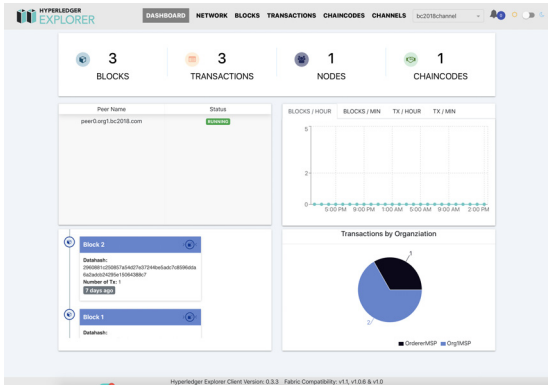
(Figure 2) Hyperledger Composer Architecture [4]

Blockchains are designed for multiparty business transactions, and organizations using blockchains are driven by transactional environments and automated processes. It works without intermediaries because it is a distributed system interworking between authorized parties through the decentralization scheme, which is one of the most useful aspects of the blockchain. To this end, the distributed blockchain ledger operates through a smart contract. Therefore, if the normal Android APK file information is registered as a blockchain, it is possible to provide a function of efficiently discriminating malicious apps in the form of forgery and fudge without a mediator.

2.3 Hyperledger Explorer

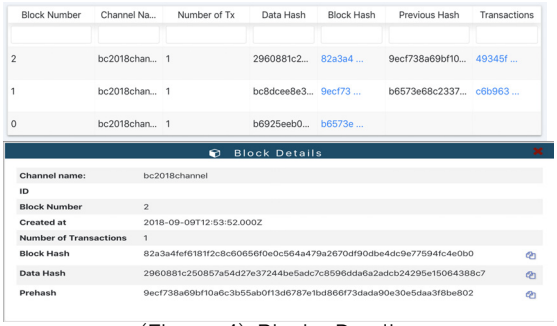
Hyperledger Fabric Blockchain Server and Web Server communicate with each other, and it is necessary to monitor information about blocks, transactions, and nodes that are created. The Hyperledger project[3] allows you to monitor blockchain networks using the Explorer. In this study, we developed the interlocking process for blockchain monitoring using Hyperledger Fabric 1.3v and Explorer.

Hyperledger Explorer uses information about the blockchain network through the PostgreSQL database (See Hyperledger Explorer Github [8] for more information). The process of linking based on Explorer is as shown in Figure 3.



(Figure 3) Hyperledger Explorer

On the main screen of Hyperledger Explorer, you can see summary information of each block, each peer and visualized information. Using the Network tab, you can see the name of the currently active peer and the Request URL (Uniform Resource Locator) to perform each requests as shown in Figure 4.



(Figure 4) Blocks Details

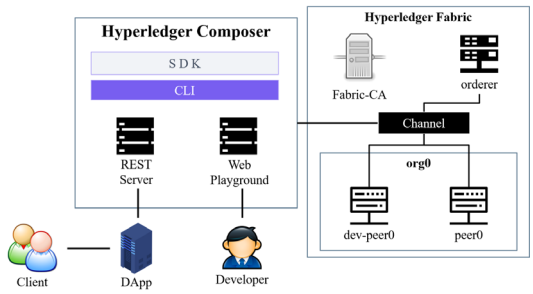
In the block tab, we can see information about the blocks that have been generated so far. It provides the block number, the name of the generated channel, the number of included transactions, various hash values, and a link to check transactions immediately. Therefore, we can provide detection procedure on legally enrolled Android APK file using the chain code block in Hyperledger Fabric framework.

3. Advanced Smart Contract for Android APK Forgery Detection

3.1 Blockchain Based Forgery Detection DApp Architecture

For implementing a blockchain based forgery detection DApp that determine fake Android mobile apps based on the extracted features from the questionable APK. In detail, we proposed an advanced smart contract from the uploaded legal APK by applying the secure hash function such as SHA-256 (Secure Hash Algorithm-256) together with additional feature set extracted from the APK to determine whether the APK is malicious/forged or normal. In this paper, smart contract for Android mobile APK file is designed using Hyperledger composer and blockchain for normal APK is constructed based on Hyperledger Fabric framework as shown in Figure 5. Using the developed DApp, the general client can judge whether the APK file is forged or not by using the blockchain framework efficiently.

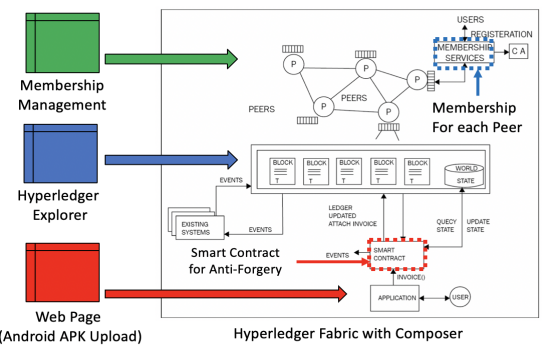
As a result, we designed forgery detection DApp that only the trust authority can (1) register the normal APK by



(Figure 5) Forgery Detection DApp

authorizing each node using the function of the consortium (private) blockchain, and (2) generate an unique one-way hash value of the APK in the apps registration process, and (3) store it in the blockchain. If it is determined that the APP is falsified, (4) the normal user nodes read the hash value of the normal APK, which is the transaction content of the blockchain, and compare it with the downloaded APK. (5) If the hash value matching the record in the blockchain is stored, it is determined as a normal apps. If the hash value differs from the hash value in the blockchain, it can be identified as a forgery apps[9]-[11].

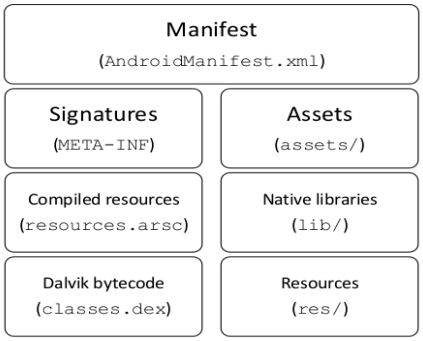
The internal system structure is composed of the following three modules as shown in Figure 6, which is based on the smart contract designed by Composer based on the Hyperledger Fabric, which can be used for forgery detection : (1) a Hyperledger Explorer that provides verification and validation of blockchain information and transactions generated inside, (2) a membership management module for granting permission to register a normal APK in a blockchain. And (3) A Web interface for uploading a corresponding APK file to check its forgery.



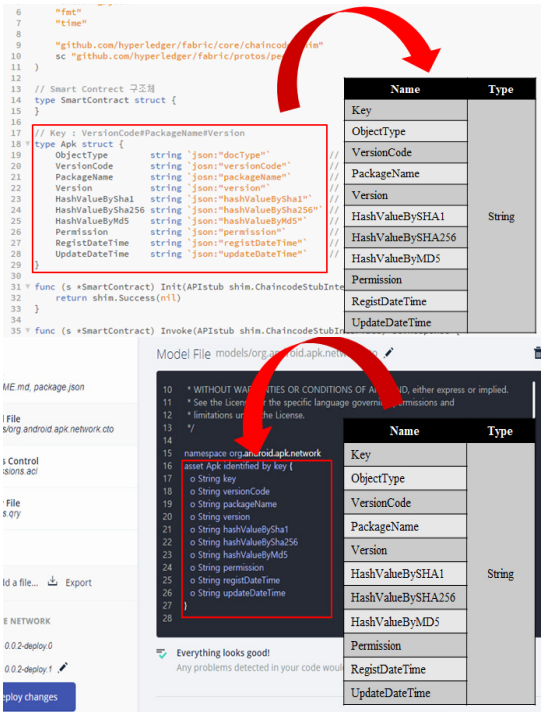
(Figure 6) Hyperledger Composed based Android Forgery Detection DApp

3.2 Smart Contract for Forgery Detection

In this study, we use the hash values together with the feature information of the APK to determine whether the APK is malicious/forged or normal based on [11]. In order to register the normal APK in the blockchain, we generate hash value from each APK and use it as reference value for determining whether it is forged or not.



(Figure 7) APK Internal Structure



(Figure 8) Smart Contract for Android APK Forgery Detection

The APK internal structure as shown in Figure 7 contains the "AndroidManifest.xml" file, and after performing a static analysis process on the APK using this, the information such as PackageNmae, VersionCode, Version, and AppName can be extracted. Therefore, in this study, after performing the static analysis on the APK and extracting the basic information of the APK, the smart contract structure as shown in Figure 8 is created as a key value and stored in the chain code.

(Key: VersionCode#PackageName#Version)

After the registration process for the normal APK file is performed based on the Smart Contract, if there is an APK to be falsified, it is necessary to upload and query the APK. Therefore, we developed Android APK Query module based on Hyperledger Composer as shown in Figure 9. In addition, web server is implemented using REST server supported by Hyperledger Composer, and it is implemented so that normal APK file can be registered as asset type in chain code in conjunction with Composer REST API.

(Figure 9) Android APK Query module based on Hyperledger Composer

3.3 Forgery Detection Procedure with Smart Contract

Transaction processing in a blockchain network is based on chain codes. When a registration request is received from each peer participating in the network, the invoke process is performed using a function called Invoke. That is, the Invoke function as shown in Figure 10 is the same as the Main function of the chain code.

In the Invoke function, initLedger is a function originally implemented for testing. Also, registApkLocally is implemented as 'Go language' to test the chain code itself. It registers the result of performing APK static analysis directly. It can link directly to perform static analysis process through Web interface.

queryApk and queryApkAll are used to inquire information about the registered APK. At this time, queryApk uses a key value to query a specific APK, and queryApkAll provides a role to inquire about all currently registered APKs.

registApk plays a role of registering a new APK, and the static analysis of the APK proceeds in the back-end part of the Website. updateApk modifies the information about the APK currently registered. verifyApk, which is a core function of the present invention, performs a blockchain-based verification process based on the hash value and the static analysis in the uploaded APK for malicious / fake application detection. At this time, when uploading a file to the Website interface, it provides a process of determining whether the APK is registered normally and a fake malicious application, and if not registered on blockchain, it returns an error code. The

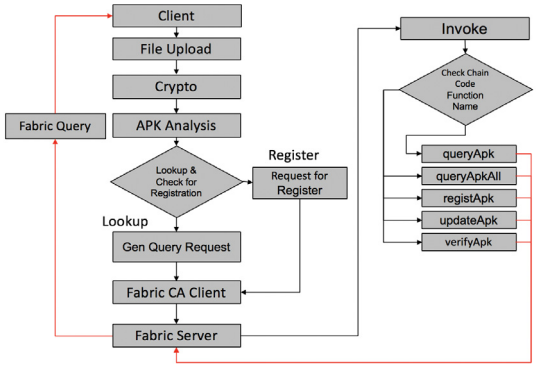
```
func (s *SmartContract) Invoke(APIstub shim.ChaincodeStubInterface) sc.Response {

    function, args := APIstub.GetFunctionAndParameters()

    if function == "initLedger" {
        // return s.initLedger(APIstub)
    } else if function == "queryApk" {
        return s.queryApk(APIstub, args)
    } else if function == "queryApkAll" {
        return s.queryApkAll(APIstub)
    } else if function == "registApk" {
        return s.registApk(APIstub, args)
    } else if function == "updateApk" {
        return s.updateApk(APIstub, args)
    } else if function == "verifyApk" {
        return s.verifyApk(APIstub, args)
    }

    // else if function == "registApkLocally" {
    //  return s.registApkLocally(APIstub, args)
    // }

    return shim.Error("Invalid Smart Contract function name.")
}
```

(Figure 10) Invoke Function

smart contract of [11] were partially used to the above contents and some modifications were made to improve the performance of Hyperledger composer-based DApp system.

The overall procedure for APK verification and registration at the site implemented in this study is as follows in Figure 11.

1. Users upload files to Web Server.
2. Apply the Crypto module to the uploaded APK to extract the hash values (SHA-1, SHA-256, MD5).
3. In addition, the APK Analysis module is run to perform static analysis to extract feature information.
4. Request the fabric chain chain server through the Fabric CA Client module to verify and register.
5. Pass the registration result to the user.



(Figure 11) Flow Chart of Normal APK Registration and Forgery APK Determination

The above-mentioned normal APK registration and forgery-and-fake APK discrimination mechanism will be described as follows in Figure 12.

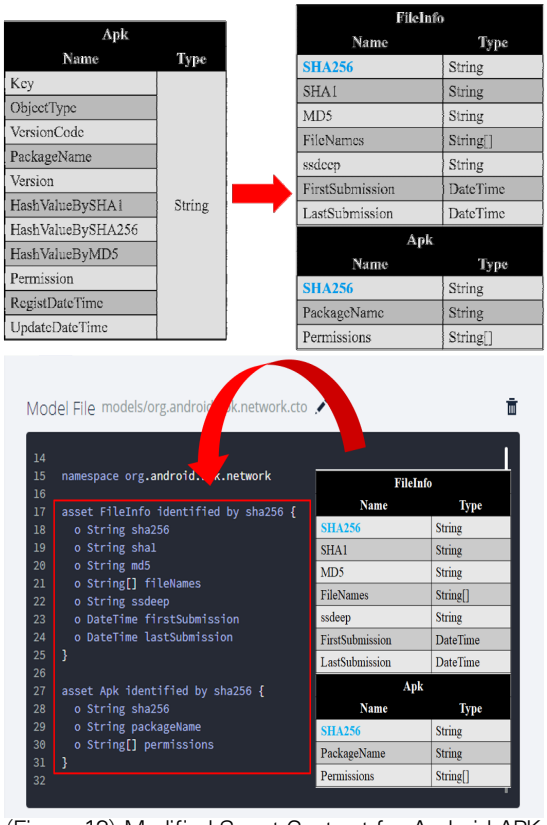
1. When APK file is uploaded for normal APK, SHA-256 one-way hash value is generated by applying Crypto module implemented in Go language.
2. Based on the extracted hash value, APK internal information such as permission, version information, used SDK information, and the like are extracted through static analysis of the APK file.
3. And then, it is checked whether the uploaded file is a registered file. If it is not previously registered, it is added as a new block inside the Hyperledger Fabric,

and the Hyper-Threading Fabric Server broadcasts the block to the entire peer.

(Figure 12) Normal APK Registration and Fake APK Discrimination Mechanism

3.4 Improvement of Smart Contract

In order to improve the efficiency of the normal APK registration and forgery and fake APK discrimination mechanism shown above, the Smart Contract is corrected / supplemented as shown in Figure 13. Key property, which is composed of 11 fields, is divided into FileInfo and Apk table, and the unnecessary fields are deleted by setting the SHA-256 value as the primary key. And we added the ssdeep field to apply Fuzzy Hashing[13,14] based similarity measurement method. As a result, we could improve efficiency in normal APK registration and forgery process.



(Figure 13) Modified Smart Contract for Android APK

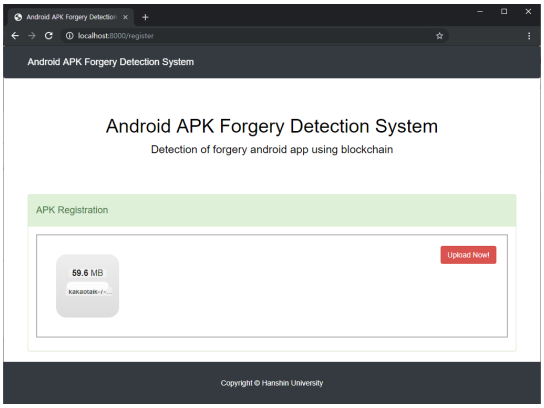
We modified the API according to the changed model and confirmed that the file name is saved as SHA-256 when uploading the APK. As a result of the implementation, it is confirmed that the APK information is normally registered in the Apk table, and that the package name and permission information are normally registered in the chain code. Therefore, the process of checking whether forgery has been performed using only the file name has been changed to a method of inquiring in the chain code using the SHA-256 hash value, and it has been confirmed that it operates normally.

4. Implementation of Forgery Android APK Detection DApp

4.1 Implementation of Web based Determination Interface

The right to access the blockchain network is only granted to authorized bodies. Therefore, it is impossible for general users to access the blockchain network. In order to control this effectively, a website is implemented so that users can perform services by providing an organization that performs a role of fake application verification and authentication processes.

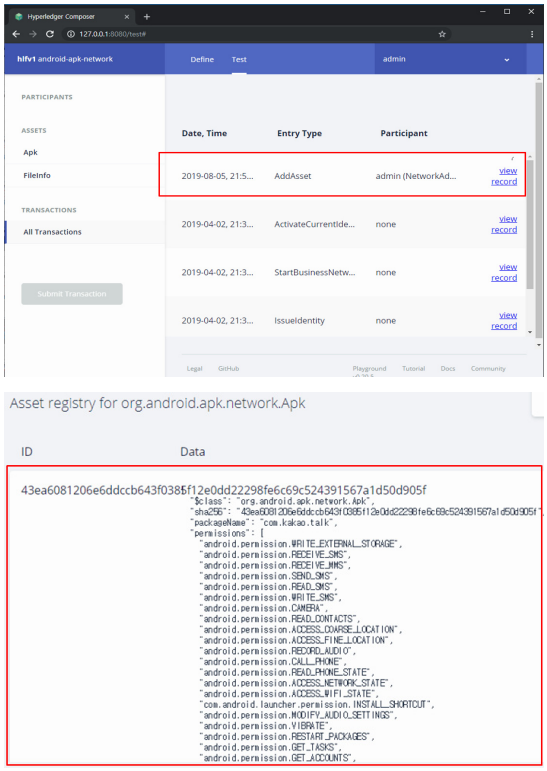
The website implemented in this paper is divided into two steps where users can directly perform the verification process on the APK file and a page where the administrator can register the new APK. Each page is implemented in a drag-and-drop manner for ease of use. The overall UI is as follow Figure 14.



(Figure 14) Android APK Forgery Detection Web Page

The user page was developed to upload the APK to the Web Server and receive the verification result for the APK, and the extracted information can be viewed through the static analysis as shown in Figure 15. The administrator page registers the static analysis result of the APK uploaded to the Web Server to the blockchain network. That is, it provides a function to register normal APK file in the blockchain through the manager page.

If registered as a normal Android APK block within the chain code, the transaction log can be checked through Hyperledger Composer. Registered chain code blocks can also be identified in APK assets. When a normal APK is registered in the blockchain, the registration time is recorded as shown in Figure 15, and the static analysis result of the registered normal app is stored.

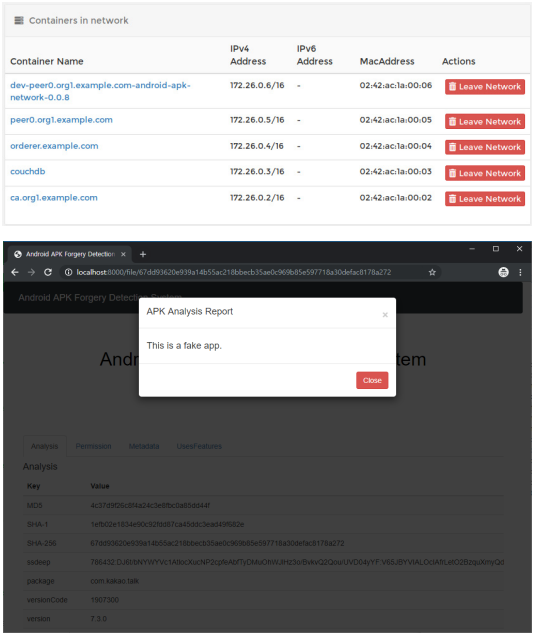


(Figure 15) APK Enrollment Transaction Log and Registered Data in Hyperledger

4.2 Forgery APK Detection Result

In recent years, blockchain technology has been applied to various fields [12]-[14]. After recording and storing the normal App in the blockchain, it is possible to determine the possibility of forgery whether or not it is included in the blockchain. We examined whether the system implemented in this study distinguishes normal and fake apps from normal and fake apps. First, we experimented with a DApp implemented for a fake app that provides similar functionality to the normal com.kakao.talk APK. As a result, we provide (1) a function

to check whether it is registered in the blockchain. Also, (2) Verification process of blockchain provides a function to discriminate malicious forgery app correctly.



(Figure 16) Blockchain based Normal/Counterfeit App Detection Result

As shown in Figure 16, we implemented the function to extract permission, metaData and usesFeatures information from each normal and counterfeit malicious app. And it stores and manages those three key-feature information extracted from the normal app in the blockchain. In the case of com.kakao.talk app (as shown in Figure 16), comparing the usesFeature extracted from the normal app and the fake malicious app, it was possible to distinguish a normal app and a counterfeit app for a user requesting discrimination based on the Hyperledger blockchain DApp implemented in this study. And we can detect forgery APK with 100% probability on experimented 25 kinds of APKs.

The administrator page registers the static analysis result of the APK uploaded to the Web Server to the blockchain network. That is, it provides a function to register normal APK file in the blockchain through the manager page. Finally, the blockchain based Android APK forgery detection system is as follow Figure 16.

5. Conclusions

In order to improve the efficiency of the normal APK registration and forgery and fake APK discrimination mechanism, Smart Contract is supplemented by setting the SHA-256 value as the primary key. And we added the ssdeep field to apply Fuzzy Hashing [15,16] based similarity measurement method. As a result, we could improve efficiency in normal APK registration and forgery process.

We implemented the function to extract permission, metaData and usesFeatures information from each normal and counterfeit malicious app. And it stores and manages those three key-feature information extracted from the normal app in the blockchain. Therefore, it was possible to distinguish a normal app and a counterfeit app for a user requesting discrimination based on the Hyperledger Fabric based detection DApp implemented with Hyperledger Composer with 100% probability on experimented 25 kinds of APKs. With proposed DApp, we can prevent the forgery and modification of the app from being installed on the user's Smartphone. In the future, the system developed in this study is expected to provide a safer environment when using smartphones.

Reference

[1] Wikipedia contributors, "Blockchain," Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=Blockchain&oldid=912153874 (accessed August 26, 2019).

[2] Commexting the World to Crypto. The World's Fastest Crypto Exchange, Blockchain Web site: https://www.blockchain.com

[3] The Linux Foundation Project, Hyperledger Project URL: https://www.hyperledger.org

[4] Welcome to Hyperledger Composer, https://hyperledger.github.io/composer/v0.19/introduction/introduction.html

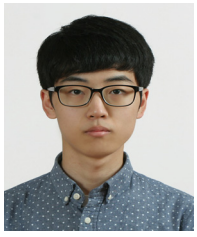
[5] Hyperledger Fabric, Accessed on Jun. 18, 2019. [Online] Available: https://hyperledger-fabric.readthedocs.io/en/release-1.3/

[6] Make your blockchain smart contracts smarter with business rules,

https://www.ibm.com/developerworks/library/mw-1708-mery-blockchain/1708-mery.html
[7] K. Christidis and M. Devetsikiotis, ‘‘Blockchains and smart contracts for the Internet of Things,’’ IEEE Access, Vol. 4, pp. 2292 – 2303, 2016.
[8] Hyperledger Explorer GitHub, https://github.com/hyperledger/blockchain-explorer
[9] Yilin Ye, Lifa Wu, Zheng Hong and Kangyu Huang, “A Risk Classification Based Approach for Android Malware Detection,” KSII Transactions on Internet and Information Systems, Vol.11, No.2, pp.959-981, 2017. https://doi.org/10.3837/tiis.2017.02.018
[10] Jingjing Gu, Binglin Sun, Xiaojiang Du, Jun Wang, Yi Zhuang, Ziwang Wang, “Consortium Blockchain-based Malware Detection in Mobile Device,” IEEE Access Vol. 6, pp(99):1-1, 2018.
[11] S. Hwang, H-W Lee, “Identification of Counterfeit Android Malware Apps using Hyperledger Fabric Blockchain,” Journal of Internet Computing and Services(JICS), Vol.20, No.2, pp.61-68, 2019. http://dx.doi.org/10.7472/jksii.2019.20.2.61
[12] Jingting Xue, Chunxiang Xu and Yuan Zhang, “Private Blockchain-Based Secure Access Control for Smart Home Systems,” KSII Transactions on Internet and Information Systems, Vol.12, No.12, pp.6057-6078, 2018. https://doi.org/10.3837/tiis.2018.12.024
[13] Jiao Li, Gongqian Liang and Tianshi Liu, “A Novel Multi-Link Integrated Factor Algorithm Considering Node Trust Degree for Blockchain-based Communication,” KSII Transactions on Internet and Information Systems, Vol. 11, No. 8, pp.3766-3788, 2017. https://doi.org/10.3837/tiis.2017.08.001
[14] Xiaojian He, Ximeng Chen and Kangzi Li, “A Decentralized and Non-reversible Traceability System for Storing Commodity Data,” KSII Transactions on Internet and Information Systems, Vol.13, No.2, pp.619-634, 2019. https://doi.org/10.3837/tiis.2019.02.008
[15] Ssdeep – Fuzzy hashing program, https://ssdeep-project.github.io/ssdeep/index.html
[16] Jesse Kornblum, “Fuzzy Hashing” [Online] Available : http://jessekornblum.com/presentations/htcia06.pdf

◐ 저 자 소 개 ◑

이 형 우(Hyung-Woo Lee)
1994년 고려대학교 컴퓨터학과(이학사)
1996년 고려대학교 대학원 컴퓨터학과(이학석사)
1999년 고려대학교 대학원 컴퓨터학과(이학박사)
2003년~현재 한신대학교 컴퓨터공학부 교수
관심분야 : 모바일/네트워크 보안, 디지털포렌식, 정보보호, 블록체인, etc.
E-mail : hwlee@hs.ac.kr



이 한 성(Han Seong Lee)
2017년 2월 한신대학교 컴퓨터공학부 졸업
2018년 9월~현재: 한신대학교 일반대학원 컴퓨터공학과 석사과정
관심분야 : 블록체인 기술, 모바일 보안, 디지털포렌식
E-mail : jkhanseong@naver.com