Large-scale Android malware detection by integrating Blockchain and IPFS for secure virus signature distribution

DOI: https://doi.org/10.54654/isj.v1i24.1085

Nguyen Tan Cam*, Pham Nhat Duy, Hoang Mai Thien Phuc

***Abstract*—The growing threat of Android malware underscores the limitations of centralized antivirus systems, which face challenges such as latency, single points of failure, and susceptibility to attacks. To address these issues, this paper introduces a decentralized framework leveraging blockchain technology via Hyperledger Fabric and the InterPlanetary File System (IPFS). The system, HypatiaX, provides secure, efficient, and transparent virus signature distribution while ensuring scalable and resilient data storage. By utilizing blockchain for virus signature management and IPFS for decentralized storage, HypatiaX supports real-time updates in distributed environment. Performance evaluations reveal low resource consumption, near-instantaneous query responses, and efficient virus scanning under diverse conditions. Advanced components, including a ledger controller, signature crawler, key manager, and IPFS client, further strengthen decentralized storage, secure key management, and automatic signature updates. This framework demonstrates significant improvements in combating Android malware while addressing the inherent flaws of traditional antivirus solutions.**

***Tóm tắt*— Mối đe dọa ngày càng gia tăng của mã độc Android nhấn mạnh những hạn chế của các hệ thống antivirus tập trung, vốn phải đối mặt với các thách thức như độ trễ, điểm lỗi đơn lẻ và dễ bị tấn công. Để giải quyết các vấn đề này, bài báo này đề xuất một khung phân phối chữ ký virus, tên HypatiaX, tận dụng công nghệ blockchain thông qua Hyperledger Fabric và InterPlanetary File System (IPFS). Hệ thống HypatiaX cung cấp khả năng phân phối chữ ký virus an toàn, hiệu quả và minh bạch, đồng thời đảm bảo lưu trữ dữ liệu mở rộng và linh hoạt. Bằng cách sử dụng Blockchain để quản lý chữ ký virus và IPFS để lưu trữ phi tập trung, HypatiaX hỗ trợ cập nhật dữ liệu chữ ký mã độc trong môi trường phân tán. Đánh giá hiệu suất cho thấy mức tiêu thụ tài nguyên thấp, phản hồi truy vấn nhanh và quét mã độc hiệu quả trong các điều kiện khác nhau. Các thành phần nâng cao, bao gồm bộ điều khiển sổ cái (ledger controller), trình thu thập chữ ký (signature crawler), trình quản lý khóa (key manager) và IPFS client, giúp tăng cường lưu trữ phi tập trung, quản lý khóa an toàn và cập nhật chữ ký tự động. Khung này mang lại những đóng góp tiềm năng trong việc chống lại mã độc Android, đồng thời khắc phục những điểm yếu vốn có của các giải pháp antivirus truyền thống.**

Keywords*— *Virus; signature, blockchain, Android malware; IPFS.

Từ khóa*— *Vi-rút, chữ ký, chuỗi khối, mã độc Android, IPFS.

I. INTRODUCTION

The rapid proliferation of malware targeting Android devices has created significant security challenges. Ensuring robust device security is increasingly critical, particularly as threats continue to evolve in complexity and spread at unprecedented speeds. Traditional antivirus systems rely primarily on signature-based scanning methods to detect and neutralize malware. However, these systems are often centralized and depend on a single entity to manage and distribute virus signatures. This centralized approach presents several vulnerabilities, including single points of failure, exposure to supply chain attacks, and delays in response time. Such delays can allow malware to propagate before the necessary signatures become widely available.

This manuscript was received on February 10, 2025. It was reviewed on March 28, 2025, revised on April 14, 2025 and accepted on May 5, 2025.

* Corresponding author

To address these limitations, temporary solutions like duplicating virus signature databases have been explored. However, these methods lack the efficiency, scalability, and security required to meet the demands of today's rapidly evolving malware landscape. There is, therefore, an urgent need for a decentralized technology that can streamline virus signature distribution while ensuring data immutability, transparency, and resilience against tampering. Blockchain technology offers a promising solution, providing a distributed ledger system where transactions are securely recorded and verified by multiple participants. This approach enhances data integrity and reliability while introducing a transparent, tamper-resistant framework for virus signature management.

Furthermore, multiple antivirus vendors and security organizations contributing virus signatures require a transparent and trustworthy platform to ensure the integrity and accountability of their contributions. A blockchain-based system facilitates this by providing a verifiable and tamper-proof ledger of submissions. This collaborative and transparent contribution model not only increases the diversity and quality of the global virus signature database, but also benefits end users by improving the effectiveness and speed of malware detection across devices and ecosystems.

Complementing blockchain, the IPFS offers efficient and resilient data storage capabilities. Unlike traditional server-based storage solutions, IPFS leverages a peer-to-peer network to distribute files, ensuring virus signatures remain accessible even if some nodes become unavailable. By integrating IPFS for the storage and distribution of virus signatures, the system achieves high availability and scalability, mitigating risks associated with centralized storage. Together, blockchain and IPFS form a robust decentralized infrastructure that is secure, efficient, and adaptable to the growing demands of malware detection.

This study contributes to the field of cybersecurity by proposing an integrated system that combines the strengths of blockchain and IPFS to enhance virus signature distribution for Android malware scanners. The primary contributions of this work include:

- Integrating blockchain and IPFS layers with an Android antivirus application.

- Conducting performance evaluations to assess the feasibility and advantages of this decentralized approach.

By leveraging the combined capabilities of blockchain and IPFS, this study aims to lay the foundation for future developments in secure, scalable, and efficient virus signature management, fostering a safer digital environment.

The remainder of this paper is structured as follows: Section 2 reviews related research on leveraging blockchain and associated technologies to address malware challenges. Section 3 covers the foundational concepts, including blockchain, the specific type of blockchain utilized, IPFS, and virus signatures. Section 4 outlines the architecture of the proposed system. Section 5 details the experimental set-up and presents the results of the study. Lastly, Section 6 concludes the paper by summarizing the findings and discussing potential directions for future work.

II. RELATED WORK

Blockchain technology has emerged as a transformative solution for enhancing malware detection, signature distribution, and ransomware mitigation, offering a decentralized and tamper-resistant architecture that addresses critical security and scalability challenges. This section presents the work related to Blockchain-based security.

Huang et al. [1] introduced a secure file-sharing system combining blockchain with the IPFS to overcome storage limitations. The system efficiently handles large documents while ensuring data confidentiality and integrity through blockchain-based access control. Similarly, Marhane et al. [2] proposed a secure data-sharing system for universities, using Hyperledger Fabric and IPFS to securely manage sensitive student information, ensuring compliance with privacy regulations.

Focusing on malware detection, Milazzo et al. [3] utilized distributed ledger technology to authenticate and distribute virus signatures. Their approach enables decentralized management of trusted virus signatures, accelerating updates while reducing dependency on single authorities. Expanding on this, Alsaiary and Ahmed [4] developed "GotchaMalware", a decentralized application using Ethereum blockchain to securely store and share malware file hashes, effectively improving detection accuracy and system resilience.

Gupta et al. [5] proposed a dual-engine system that integrates signature-based and multi-antivirus detection strategies. Using blockchain for collaborative evaluation of Portable Executable (PE) files, their framework ensures secure and tamper-proof storage while demonstrating significant improvements in detection speed and accuracy. Fuji et al. [6] extended blockchain-based malware signature sharing, achieving a 4% reduction in false negatives and a 2.5% reduction in false positives compared to heuristic-based methods, despite variability in accuracy among users.

Further emphasizing secure malware detection, Rahman et al. [7] introduced a tamper-proof malware signature sharing tool leveraging blockchain's robustness. Robert et al. [8] enhanced malware detection with a dual private blockchain system that integrates static and dynamic analysis for smartphone applications, improving accuracy and dynamic resource monitoring.

Targeting Android malware, Boobalan et al. [9] proposed a Multi-Feature Model (MFM) framework using Ethereum blockchain to combine static, dynamic, and prestatic analysis. Their approach demonstrated high accuracy and efficiency on the CICAndMal2017 dataset.

Anas Kwefati [10] introduces HuntChain, a decentralized malware detection tool designed to address the limitations of centralized antivirus databases. By utilizing blockchain technology, HuntChain integrates signature-based detection methods such as "Top-and-Tail" and hash-checking, storing malware data on an immutable blockchain to enhance security and transparency. Evaluations demonstrate their robustness and accessibility as an open-source solution for malware detection and prevention.

Hu et al. [11] propose a blockchain-based ecosystem for auditing software security, featuring interconnected networks for software verification, malware reporting, and public announcements. The framework incorporates distributed voting, consensus protocols, and pseudonymous auditing, reducing reliance on centralized systems while rewarding participants with virtual credit points. This approach improves credibility and strengthens malware prevention mechanisms.

Khellaf and Boudouda [12] present a blockchain-enhanced Mobile Application Management (MAM+) system aimed at securing mobile enterprise environments. By integrating mobile agents, blockchain, and a decentralized authentication system, MAM+ autonomously detects illegitimate activities and facilitates real-time responses. Key features include secure data transfer, a blacklist for threats, and a whitelist for trusted applications, ensuring integrity and transparency in enterprise security.

Rohith and Kaur [13] provide a comprehensive review of malware detection techniques in antivirus software, emphasizing signature-based detection. This method compares files against a database of known virus signatures to identify malicious files, showcasing its continued relevance in cybersecurity.

Deok Gyu Lee [14] proposes a blockchain-based system integrated with deep learning for malware detection. Malware is converted into grayscale images, registered on the blockchain, and analyzed using neural networks. This decentralized approach improves real-time data processing and enhances security while overcoming challenges like slow processing due to blockchain size. Similarly, Denysiuk et al. [15] integrate blockchain with deep learning by utilizing a decentralized network of subnets for collaborative code analysis. Their Proof-of-Action (PoA) consensus mechanism validates

detection results, achieving high accuracy (98.81–99.33%) across various malware classes.

Kumar et al. [16] extend the hybrid approach by combining blockchain with federated learning for Android malware detection in IoT environments. The framework employs Local Neural Network (LNN) models to aggregate static and dynamic features, with smart contracts that ensure the authentication and aggregation of secure models. The system effectively reduces computational costs and improves detection performance through decentralized training and IPFS-based data storage.

Martin et al. [17] propose a consortium blockchain to consolidate malware signatures, improving detection rates and reducing false positives. Gu et al. [18] present the CB-MDEE framework, which builds a multi-feature model to accurately classify Android malware, leveraging blockchain to enhance evidence collection and security. Sheela et al. [19] combine signature-based and behavioral-based detection methods within a blockchain system, achieving high detection rates with low false positives.

Cui et al. [20] introduce a blockchain-based app marketplace model to detect potentially unwanted apps (PUAs) in Android marketplaces. By combining metadata mining, sentiment analysis, and runtime behavior analysis, the system enhances marketplace security and user trust. Gupta et al. [21] develop a distributed intrusion detection framework powered by blockchain, integrating signature-based, behaviorbased, and multi-antivirus detection engines to classify files as benign or malicious. Their system uses blockchain to collaboratively evaluate executable files and store results securely.

Wressnegger et al. [22] investigated the vulnerabilities in signature-based malware detection, introducing the concept of antivirus-assisted attacks. They proposed an automated method for reverse-engineering malware signatures from commercial antivirus products, revealing that many signatures are overly simplistic and lack contextual constraints. This weakness allows attackers to create “malicious markers”, benign data embedded with these signatures to generate false positives. The study demonstrated the practicality of such attacks in scenarios like masking password guessing, email deletion, and browser cookie removal. The authors highlight the importance of incorporating context and semantics into signature-based detection mechanisms to mitigate these vulnerabilities.

Senanayake et al. [23] introduce Defendroid, a tool combining blockchain-based federated neural networks with Explainable Artificial Intelligence (XAI) to detect code vulnerabilities in Android applications. This real-time system, integrated into Android Studio, preserves data privacy while enabling collaborative model training. Evaluations highlight its effectiveness, achieving a 96% accuracy rate and an F1-score for binary and multi-class vulnerability detection.

Park et al. [24] examined ransomware detection in IoT environments, integrating blockchain with AI and Software-Defined Networking (SDN) to mitigate threats and address IoT-specific challenges.

Kalphana et al. [25] introduced a hybrid system that combines a deep learning model with a cryptographic approach for detecting and mitigating Android ransomware threats. The methodology involves preprocessing APK files to extract relevant features using Squirrel Search Optimization (SSO), followed by classification using an adaptive deep saliency AlexNet model. The identified benign data is securely stored in the cloud using a hybrid cryptographic approach that integrates Homomorphic Elliptic Curve Cryptography (ECC) and Blowfish encryption. The experimental results demonstrate the model’s high detection accuracy of 99.89%, outperforming traditional deep learning and cryptography models.

Although a growing number of studies have explored the integration of blockchain and IPFS in areas such as secure data sharing, malware detection, and collaborative antivirus systems, the combination of Hyperledger Fabric and IPFS specifically for managing virus signature

databases remains limited. Most prior works focus on Ethereum-based implementations or general-purpose blockchain platforms, with few addressing enterprise-grade frameworks like Hyperledger Fabric. This highlights a notable research gap, especially in building a scalable and secure infrastructure for decentralized virus signature distribution and verification using permissioned blockchain systems. The proposed work aims to fill this gap by leveraging Hyperledger Fabric's robust access control and IPFS's decentralized storage to enhance trust, integrity, and efficiency in malware signature management.

III. Background

A. Blockchain

Blockchain is an immutable digital record-keeping system that simplifies tracking transactions and assets within a business network. These assets can range from physical items to intangible assets like intellectual property. By recording nearly all exchanges on a blockchain network, participants benefit from reduced risk and lower costs.

Blockchain has three main components are: Distributed Ledger Technology (DLT), Immutable Record and Smart Contract.

DLT: DLT allows all participants within a network to access a shared, unalterable record of transactions. Because the ledger is shared, each transaction is recorded only once, eliminating the duplication often seen in traditional business networks.

Immutable Record: Once a transaction is entered on the shared ledger, it becomes unchangeable and tamper-proof. If an error occurs, a corrective transaction is added, with both the original and corrective entries remaining visible, ensuring transparency and trust.

Smart Contract: These subprograms, stored on the blockchain, execute automatically when predefined conditions in the contract are met. Smart contracts streamline transactions by enforcing established rules, such as those governing bond transfers or insurance payouts.

Blockchain operates through a systematic and secure process that ensures the integrity and reliability of recorded data. Each transaction is stored as a unique block of data containing detailed information about the asset's movement. This information may include who was involved, what was transacted, the time and location of the transaction, the amount involved, and even specific conditions of the asset, such as the temperature of a food shipment. These blocks are then linked sequentially to form a chain, with each block connected to the ones before and after it. This chaining process creates a complete timeline of the asset's journey or changes in ownership, confirming the exact timing and sequence of all transactions. The design ensures transparency and accountability throughout the system.

Security is a fundamental aspect of blockchain. The blocks are securely connected, making it impossible to alter or insert a block without affecting the entire chain. This tamper-resistant design safeguards data integrity, ensuring that the records remain accurate and trustworthy. The result of this process is a secure, irreversible ledger known as a blockchain. As new blocks are added, they reinforce the validity of previous blocks, making the entire blockchain transparent and unchangeable. This robust structure is a cornerstone of blockchain technology, enabling it to provide reliable and decentralized solutions across various industries.

B. Hyperledger Fabric

Hyperledger Fabric [26] is a permissioned blockchain platform designed specifically for enterprise use, offering several key advantages over public blockchains like Bitcoin or Ethereum. Its modular architecture provides flexibility and customization for various industries. Fabric supports smart contracts, or "chaincode", written in commonlyused programming languages like Java, Go, and NodeJS, making adoption easier for enterprises. The platform places a high priority on privacy and security, employing a channel architecture and private data features that ensure sensitive information is shared only with authorized participants. Unlike open blockchains that rely

on resource-intensive consensus mechanisms (such as Proof of Work) due to their anonymous, trustless nature, Hyperledger Fabric operates within a network of known participants, allowing for more efficient consensus protocols.

Fabric's modularity also extends to critical components like ordering services, membership providers, and consensus mechanisms, offering flexibility in configuration to meet diverse enterprise needs. The pluggable consensus mechanism accommodates different trust models, supporting both fault-tolerant and Byzantine fault-tolerant ordering services.

In this study, Hyperledger Fabric was selected over Ethereum due to its enterprise-oriented architecture, modular design, and flexible permissioned network structure for managing and distributing virus signatures in a secure, efficient, and controlled environment.

Unlike Ethereum, which operates as a public, permissionless blockchain, Hyperledger Fabric supports permissioned networks, allowing only verified and authenticated entities (such as antivirus vendors, researchers, or administrators) to participate. This characteristic is essential for maintaining data integrity, ensuring trust among contributors, and preventing unauthorized access or malicious uploads of virus signatures.

From a performance perspective, Fabric supports high throughput and low latency by using pluggable consensus protocols, which are better suited for closed networks compared to Ethereum's Proof-of-Work or Proof-of-Stake mechanisms. Furthermore, Fabric's support for private channels ensures data privacy between specific organizations, which is not easily achievable in public blockchains like Ethereum.

C. InterPlanetary File System

The IPFS operates as a peer-to-peer network for storing and sharing data in a decentralized way. Instead of traditional location-based addressing, IPFS uses content addressing, where each file or piece of data is given a unique cryptographic hash based on its contents. When someone requests a file, the network finds the peers that hold copies of the data with the matching hash, retrieving it directly from them. This approach enables efficient data retrieval, as popular files can be accessed from multiple peers simultaneously, reducing the load on any single server. IPFS's decentralized design enhances data availability, reduces reliance on central servers, and improves resilience against data loss or censorship.

D. Virus Signature

A virus signature, or virus definition, is a unique string of code or a pattern that identifies a specific virus. Antivirus software uses these signatures to detect known viruses within a system, much like fingerprints are used to identify individuals. During a scan, the antivirus compares the code of files and programs against a database of virus signatures. If a match is detected, the software can quarantine or remove the malware, thereby protecting the system from potential harm.

IV. METHODOLOGY

A. System Overview

This study proposed a system, named HypatiaX. This system is designed for a collaborative antivirus ecosystem where trusted organizations (e.g., antivirus labs, research institutions,…) serve as peer nodes and certificate authorities. Their motivation is to collaboratively maintain the integrity of virus signature updates and reduce central points of failure. The distributed Android malware scanning application (HypatiaX) is shown in Figure 1. The proposed system operates through a multi-component workflow designed to enhance malware detection and signature management. At its core is the Blockchain network, which acts as a secure and decentralized repository for virus signature metadata. Supporting this, the system incorporates a Ledger Controller, Signature Crawler, Key Manager, and IPFS to enable efficient storage, retrieval, and verification of virus signature data. Additionally, a Malware Scanner ensures real-time virus detection on Android devices, providing fast and reliable access to the signature database while maintaining data integrity.

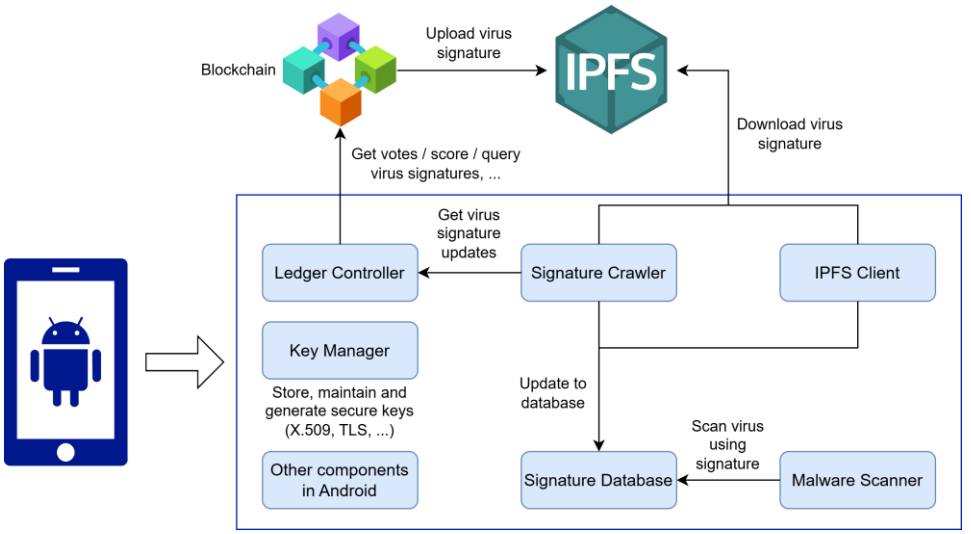

Figure 1. The architecture of HypatiaX - a blockchain-based virus signature distribution system for Android malware scanners

B. System Components

This section details the function of the components in the proposed system.

Blockchain: This stud utilizes Hyperledger Fabric, a robust blockchain framework that incorporates essential components including a distributed ledger, authorized organizations, shared channels, certificate authorities, smart contracts, and an ordering service. The workflow begins with virus signatures being uploaded and stored on the distributed ledger by authorized organizations through a shared channel. Certificate authorities authenticate participants, ensuring only verified entities can interact within the network. Smart contracts or chaincode enforce business rules, handling virus signature transactions and updating the ledger when necessary. The ordering service arranges transactions in sequence, ensuring consistent data across the network. Android applications then connect to the blockchain to retrieve the latest virus signatures securely, using them to detect and prevent malware.

IPFS: Manage decentralized storage and retrieval of virus signatures. Since files on IPFS are immutable, modifications to the virus signature database are handled by merging the updated signatures with the previous data into a new file, which is then uploaded as a new object to IPFS. The updated file receives a new content identifier (CID), and this CID is then shared with the administrator and recorded on the blockchain.

Ledger Controller: Coordinates and manages actions related to the ledger. This layer manages interactions with the Hyperledger Fabric blockchain network by establishing secure gRPC (gRPC Remote Procedure Call) connections to peer nodes within the organization. It handles identities and cryptographic keys, and performs various ledger operations, such as retrieving virus signature votes from organizations, calculating scores, and querying signatures. The layer leverages the KeyManager to securely manage keys and runs a background job to execute blockchain transactions, updating the user interface with progress messages as needed. It also includes utility methods for JSON formatting, file manipulation, and string conversion to facilitate certificate handling and data persistence. In essence, LedgerController orchestrates data retrieval from the blockchain via a peer node within the organization.

Signature Crawler: Collects information from the antivirus ledger to identify updates for new virus signatures. This layer is responsible for managing signature updates and downloads. It can be configured to automatically check for new updates in the background by interacting

with the LedgerController layer and notifying the user when updates are available. Once a new signature update is detected and the user confirms, the layer initiates the download through the IPFS layer.

Key Manager: Securely store, maintain, and generate cryptographic keys used in various tasks. This layer leverages Android's Keystore to manage identities securely. Utilizing Android's hardware-backed keystore is the best practice for applications that handle sensitive data and require strong security measures to protect cryptographic keys and operations. In this study, the method manages the following three types of keys:

- X.509 Certificate: Serves as the user's digital identity within the organization. The application prompts the user to select an identity certificate, which is pre-installed in the Android Keystore by the user.

- Organization's Private Key: Enables secure, authenticated, and authorized interactions within the network. In Android, the user's installed identity certificate includes a public certificate and a password-protected private key within a file. This key is accessible once the X.509 identity certificate is installed in the Android Keystore.

- Transport Layer Security (TLS) Certificate: Used by the peer node with which the proposed system connects for secure communications. It ensures that all communications are encrypted and that all parties are authenticated and trusted. This certificate can either be installed in the Android Keystore or hard-coded into the application.

IPFS Client: This client layer interacts with IPFS to download distributed virus signatures. IPFS can be configured to run on localhost for increased decentralization or on a server node within the organization. This layer is invoked by the SignatureDatabase class to download virus signatures.

Signature Database: Stores malware signatures used for scanning. This study uses ClamAV [27] style signatures.

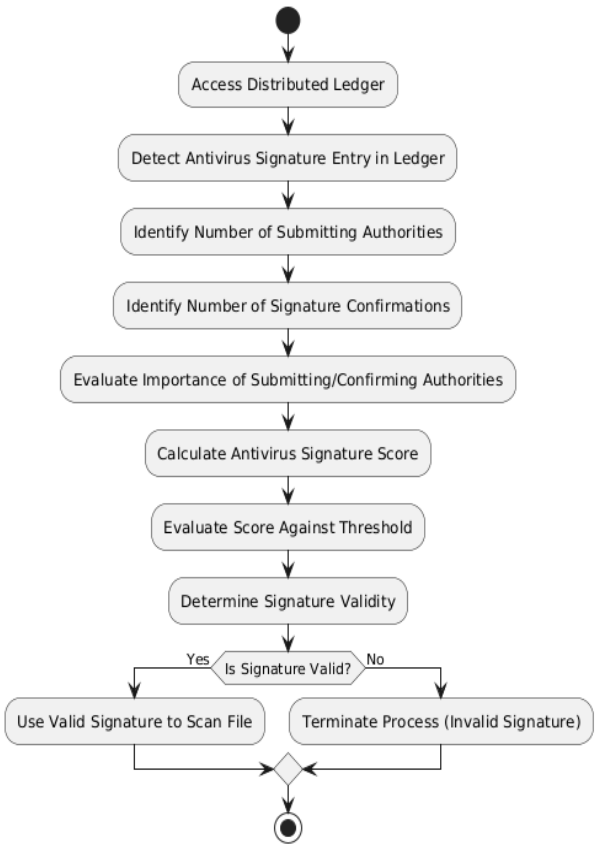

Figure 2. Antivirus signature verification and usage workflow

Malware Scanner: Real-time malware scanning for devices by monitoring file system changes and leveraging multi-threading for efficient scanning.

Other components in Android: Other components that provide essential functions for the device to operate, such as the Operating System, Network Interface, etc.

C. Main workflows of the proposed system

1. Antivirus signature verification and usage workflow

This process (Figure 2) outlines the steps taken to verify and utilize an antivirus signature retrieved from a distributed ledger:

- Step 1: Access distributed ledger: The system begins by accessing the decentralized, tamper-resistant ledger shared across multiple nodes.

- Step 2: Detect antivirus signature entry: It scans the ledger to detect entries related to antivirus signatures.

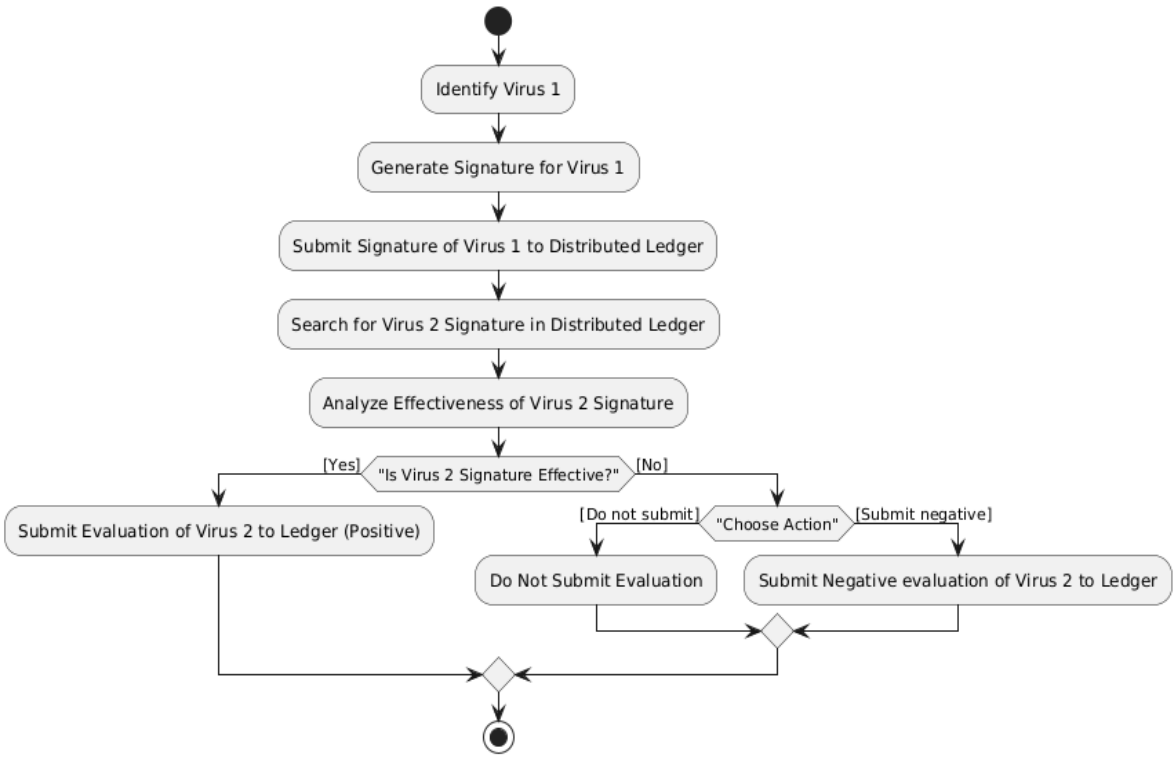

Figure 3. The process by which a signature authority creates, submits, and evaluates a virus signature to a distributed ledger

- Step 3: Identify submitting authorities: The system identifies how many trusted authorities have submitted the specific signature.

- Step 4: Identify signature confirmations: It then checks how many confirmations or validations that signature has received from peer organizations.

- Step 5: Evaluate authority importance: The system assesses the importance or credibility of each authority involved in submission and confirmation. This may depend on reputation, past accuracy, or domain expertise.

- Step 6: Calculate antivirus signature score: A score is calculated based on submission count, number of confirmations, and the weight of each authority involved.

- Step 7: Evaluate score against threshold: The computed score is compared against a predefined trust threshold.

- Step 8: Determine signature validity: Based on the comparison, the system determines whether the signature is valid or not.

- Step 9: Conditional branch:

If the signature is invalid, the process is terminated to ensure safety.

If the signature is valid, the system proceeds to the next step.

- Step 10: Scan file using antivirus signature: The validated signature is used to scan the user's file for potential malware threats.

2. Signature management process workflow

This process (Figure 3) outlines how antivirus signatures are created, submitted, and evaluated within a distributed ledger system. It begins with the identification of a new virus (Virus 1), for which a unique signature is generated and submitted to the distributed ledger. Subsequently, the system searches for the signature of another virus (Virus 2) in the ledger and analyzes its effectiveness.

If the Virus 2 signature is deemed effective, a positive endorsement is submitted to the distributed ledger, confirming the signature's reliability. However, if it is found to be ineffective, the process branches into two possible actions: either no endorsement is submitted, or a negative endorsement is issued to signal the signature's unreliability. This mechanism ensures that only validated and trustworthy antivirus signatures are promoted within the network.

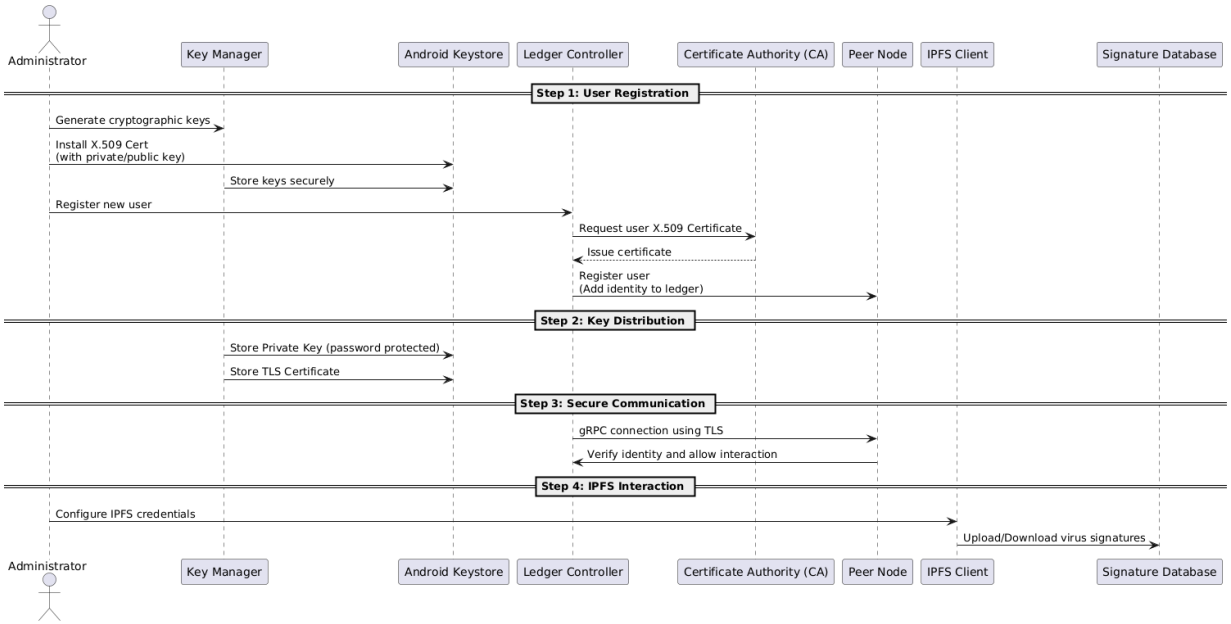

Figure 4. User Registration, Blockchain Interaction, and Key Distribution

3. User registration, Blockchain interaction, and Key distribution workflow

In the HypatiaX system, the user registration process begins with the administrator generating and installing an X.509 identity certificate into the Android Keystore via the Key Manager. This certificate serves as the user's digital identity and is securely protected within the device's hardware-backed keystore. (Figure 4).

The administrator then uses the Ledger Controller to initiate the user registration request to the Certificate Authority (CA), which is responsible for issuing the X.509 certificates. Once issued, the Ledger Controller communicates with the Peer Node to record the user's identity into the distributed ledger, ensuring that it is verifiable and securely linked to the corresponding cryptographic keys.

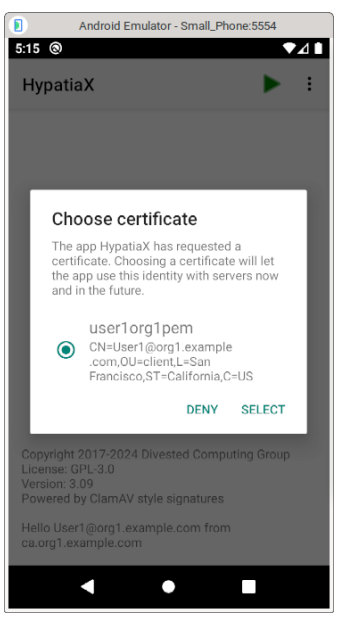

Figure 5. Application requires PKCS 12 certificate when using blockchain feature

Following registration, the Key Manager manages three key types: (1) the X.509 certificate for identity, (2) the private key for secure authentication, and (3) the TLS certificate used to establish encrypted communication with the blockchain's peer node. Communication between HypatiaX and the blockchain network uses secure gRPC channels protected by TLS.

Once the user is successfully authenticated, the HypatiaX application can interact with IPFS to upload or download virus signature data. The IPFS Client accesses the Signature Database to retrieve or store signatures, ensuring that updates are conducted in a secure and trustworthy manner.

V. Experiment and Discussion

A. System Configuration

The experiments scenarios in this study were performed on a personal computer with detailed configurations as shown in Table I.

TABLE I. SYSTEM CONFIGURATION

Component	Value
Operating System	Linux Mint 21.3 x86 64
Kernel	5.15.0-112
CPU	Intel i7-2760QM
Memory	23907 MiB

B. Result

To use the application, two types of certificates need to be installed on the Android device: a TLS certificate and a PKCS (Public Key Cryptography Standards) 12 certificate (containing the user's public and private keys). Additionally, the peer node address and the server address used for TLS certificate authentication must be provided to establish a secure connection. This peer node must belong to the same organization that issued the PKCS 12 certificate to the user. The application will then establish a gRPC connection to this peer node. Figure 5 shows an example of requesting a PKCS 12 certificate.

After selecting the appropriate licenses, the app can look up the virus signature on the blockchain network. If there is a signature with enough consensus organizations, the signature information will be saved. An example of an application performing a signature query on a blockchain network is shown in Figure 6.

To enable downloads from IPFS, an IPFS daemon node must be running either on localhost or on a server. With this setup, virus signatures can be downloaded, and scanning can begin. Testing with the virus sample in [28] shows that most Android viruses in APK format are effectively detected. Figure 7 shows an example of an application detecting malware and notifying the user.

C. Discussion

On Android devices, Hypatia [29] is considered an effective open-source malware scanner, leveraging a ClamAV-style signature database. It enables regular and real-time scanning of the system, internal memory, external storage, and installed applications without requiring an internet connection, except for downloading signature updates. This software is highly efficient, with minimal battery impact, fast scanning speeds, and low memory usage.

To assess the effectiveness of the proposed system, this study conducted a comparison between the original Hypatia app and the

Figure 6. An example of executing a signature query on a blockchain network

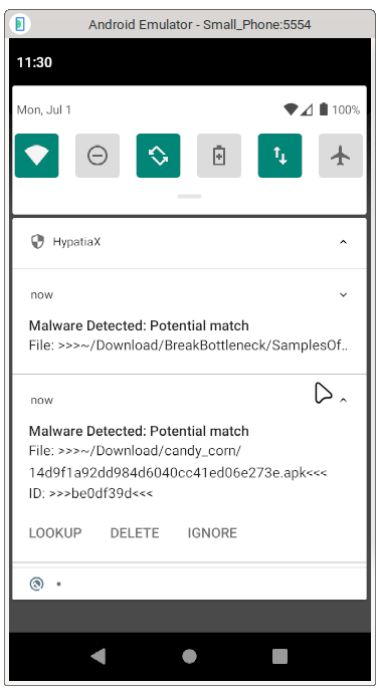

Figure 7. An example of an application detecting malware and notifying the user

enhanced HypatiaX version. The comparison in Table II highlights the key differences in functionality between the Hypatia and HypatiaX apps. Both applications share core features such as the use of a signature database (ClamAV-style) and malware scanning capabilities, ensuring a fundamental level of security. However, HypatiaX significantly extends the feature set by integrating advanced components such as a ledger controller, signature crawler, key manager, and IPFS client. These additions enhance the app's functionality by providing greater support for decentralized storage, secure key management, and automated signature updates.

TABLE II. FEATURE COMPARISON BETWEEN HYPATIA AND HYPATIAX APPS

Feature	Hypatia	HypatiaX
Signature Database	✓	✓
Malware Scanner	✓	✓
Ledger Controller	×	✓
Signature Crawler	×	✓
Key Manager	×	✓
IPFS Client	×	✓

This study deployed the proposed system and evaluated its performance. Table III summarizes the results of the blockchain network and the virus scanning application (HypatiaX) after deployment. Under ideal conditions, the blockchain network operates efficiently, consuming minimal system resources. However, in real-world scenarios, a blockchain query may take some time to return results, depending on geographic factors. The most significant bottleneck remains the human factor: much of the time is spent by researchers reviewing virus signatures and by administrators of the consensus organization in installing the chaincode. The time from signature issuance to usability also depends heavily on the number of nodes holding IPFS data and the speed of the organizations' signature consensus.

The resource consumption metrics for peer and orderer nodes show that the system operates efficiently under ideal conditions. For example, the peer node recorded a low CPU usage of 2.18% and memory consumption of 46 MiB, while the orderer node utilized only 0.11% CPU and 15 MiB memory. Network input/output metrics were also minimal, with the peer node handling 4.3 MB and 1.12 MB for input and output, respectively, and the orderer node processing significantly smaller amounts.

TABLE III. PERFORMANCE OF BLOCKCHAIN NETWORK AND HYPATIAX APPLICATION AFTER DEPLOYMENT

Component	Value
Peer Node	CPU: 2.18% ; Memory: 46MiB; Net I/O: 4.3MB / 1.12MB
Orderer Node	CPU: 0.11%; Memory: 15MiB; Net I/O: 82.1kB / 254kB
Virus signature query time from application in virtual machine	Depending on the network, almost instant in simulation
Function call time in chaincode from peer node	Depending on the network, almost instant in simulation
Chaincode	CPU: 0.00%; Memory: 5.23MiB; Net I/O: 8.31kB / 8.2kB
Virus scanning speed	Depends on the number of files; Xperia XZ1: 115s @ 95MB/s
Download speed from IPFS	Depends on network and number of nodes containing data, fast if stored on localhost node
IPFS node running localhost on Termux	Memory: 80MiB
Speed of updating new signatures in the app	Subject to network conditions
Query speed from app to blockchain network	Depends on network and nearest node location, almost instantaneous in simulated network
Try a query every 0.1s	Query node CPU usage increased by 2%, Memory increased by 30MiB
Time from signature issuance to signature being updated on the application	Depends on how fast the organization votes in favor
Time the virus signature appeared on IPFS	Depending on the network and the number of IPFS nodes containing the data

Figure 8 highlights the component differences between HypatiaX and the original Hypatia system. Components with a gray background represent the original modules of Hypatia, while the newly added modules in HypatiaX are shown with an orange background. This figure demonstrates that once the virus signature database has been downloaded to the user's device, the scanning speed of HypatiaX is independent of the server configuration used to deploy Hyperledger Fabric and IPFS. The server configuration only impacts operations such as uploading, downloading, and storing virus signatures on the IPFS and blockchain network.

Virus signature queries and chaincode function calls demonstrated near instantaneous response times in simulated environments, indicating that the blockchain network effectively supports real-time applications. Virus scanning speeds varied depending on file volume, with the Xperia XZ1 achieving a scanning rate of 95 MB/s over 115 seconds for a given dataset. The download speeds of IPFS depended on the network and the number of nodes that hosted the data, with the performance being optimal when stored locally.

The results also highlight the impact of system configurations and network conditions on performance. For example, the time required to update virus signatures on the application was influenced by the speed at which organizations reached consensus and the number of IPFS nodes containing the data. Querying the blockchain network showed increased resource usage, with a 2% rise in CPU and 30 MiB additional memory on the query node when queries were made every 0.1 seconds. These findings highlight the system's scalability and effectiveness in distributing virus signatures and detecting malware. Nonetheless, its performance still relies on network conditions and consensus latency, indicating potential areas for optimization to reduce delays in practical deployments.

In future work, it is crucial to design and conduct large-scale testing scenarios involving a high number of participating nodes to simulate

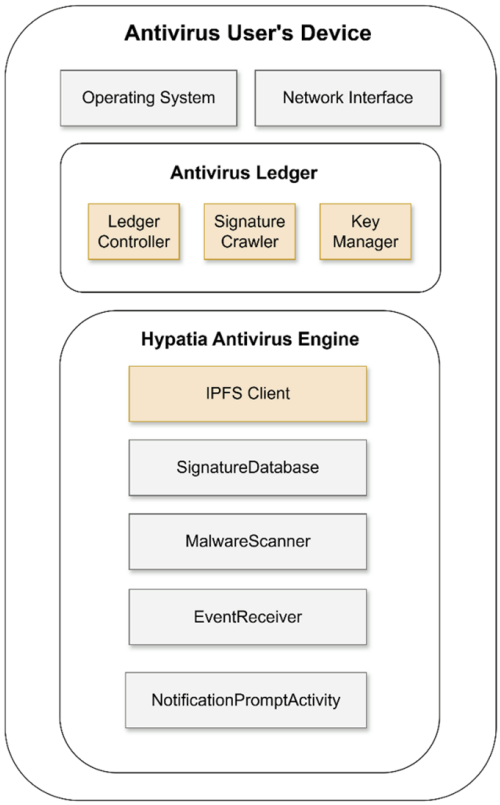

Figure 8. The component differences between HypatiaX and the original Hypatia system

real-world deployment conditions. These experiments will help evaluate how the system handles concurrent requests from multiple users who may simultaneously upload or download virus signatures. Additionally, future evaluations should include metrics such as transactions per second (TPS) and data propagation delays across the blockchain network. This is especially important to ensure that the system can deliver timely updates and maintain reliability when rapid distribution of new virus signatures is essential for user protection.

VI. Conclusion

This research proposed and implemented a Hyperledger Fabric network for virus signature scoring and distribution, utilizing IPFS for storage, with the HypatiaX application as the virus signature consumer. The findings highlight the potential of blockchain technology and IPFS to improve both the security and efficiency of virus signature distribution. This integration not only

strengthens the integrity and reliability of virus signatures through the immutability and transparency of blockchain but also enhances data storage and distribution through the decentralized file system of IPFS. Additionally, the results demonstrate that Hyperledger Fabric is a suitable choice for building a decentralized, secure, and efficient virus signature distribution system. These outcomes open opportunities for practical applications to enhance security and provide a foundation for further research and development in information security and data distribution.

Despite the positive results, this study has some limitations that warrant consideration. Due to the complexity and high configurability of the Hyperledger Fabric blockchain network, this study have not yet fully leveraged available features such as the gossip protocol, leader election, anchor peers, state database, and peer channel-based event services. Utilizing these features could optimize the architecture and expand the functionality of the blockchain network. Additionally, the study did not place sufficient focus on evaluating the performance of the Hyperledger Fabric network during deployment. Network performance significantly impacts operational smoothness, speed, and latency, which are essential for the effective distribution of antivirus signatures. A detailed analysis of scalability, transaction processing speed, and overall network performance when integrated into the virus scanning application is still needed.

In the future, we also plan to run a peer node of the blockchain network directly on Android devices. This is feasible due to Hyperledger Fabric's support for the ARM64 architecture [26]. By leveraging the capabilities of Termux, we will download and install the Hyperledger Fabric ARM64 binary, configure the node settings, and initiate a node process. This setup enables Android devices to participate in the Hyperledger Fabric network as peer nodes, expanding the accessibility and flexibility of blockchain applications and enhancing decentralization. We aslo apply the approach in this study on other domains like IoT malware detection [30, 31].

ACKNOWNLEDGMENT

This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund.

REFERENCES

[1] Huang, H.-S., Chang, T.-S., & Wu, J.-Y. (2020). A secure file sharing system based on IPFS and blockchain. *Proceedings of the 2nd International Electronics Communication Conference*, 96–100.

[2] Marhane, K., Taif, F., & Namir, A. (2023). Secure sharing of university data using Hyperledger Fabric and IPFS system. *Procedia Computer Science*, 224, 163–168. Elsevier.

[3] Milazzo, A. M., Schiatti, L., Giordano, G., & Viale, E. (2018). Antivirus signature distribution with distributed ledger. *US Patent 10,063,572*, Google Patents.

[4] Alsaiary, N. M., & Ahmed, S. (2024). Application of blockchain technology in securing mobile applications. *AIP Conference Proceedings*, 3072(1). AIP Publishing.

[5] Gupta, S., Thakur, P., Biswas, K., Kumar, S., & Singh, A. P. (2021). Toward a novel decentralized multi-malware detection engine based on blockchain technology. *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2020, Volume 2*, 811–819. Springer.

[6] Fuji, R., Usuzaki, S., Aburada, K., Yamaba, H., Katayama, T., Park, M., Shiratori, N., & Okazaki, N. (2019). Investigation on sharing signatures of suspected malware files using blockchain technology. *International Multi Conference of Engineers and Computer Scientists (IMECS)*, 94–99.

[7] Abdul Rahman, S. H., Nevin Gabriel, C., Haw, S. C., & Zainuddin, A. A. (2023). Blockchain malware detection tool based on signature technique. *Advances in Artificial Intelligence and Machine Learning*, 3(4), 1654–1670. Shimur Publications.

[8] Robert, P., Senkamalavalli, R., Vedanarayanan, V., & Manivannan, D. (2023). Blockchain-based malware detection system for smartphone applications. *2023 8th International Conference on Communication and Electronics Systems (ICCES)*, 216–221. IEEE.

[9] Boobalan, P., Keerthana, R., Nandhini, K., & Vignesh, P. (2020). Multi feature detection and signature sharing of Android malware using blockchain. *IIRJET*, 5(3).

[10] Kwefati, A. (2021). HuntChain Project: A

blockchain-based malware detection tool.

[11] Hu, Q., Asghar, M. R., & Zeadally, S. (2021). Blockchain-based public ecosystem for auditing security of software applications. *Computing*, 103(11), 2643–2665. Springer.

[12] Khellaf, R., & Boudouda, S. (2024). Enhancing mobile enterprise security: A blockchain and agent paradigm-based approach for continuous protection and rapid adaptation. *IEEE Access*. IEEE.

[13] Rohith, C., & Kaur, G. (2021). A comprehensive study on malware detection and prevention techniques used by anti-virus. *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, 429–434. IEEE.

[14] Lee, D. G. (2021). A study on malicious code detection using blockchain and deep learning. *KIPS Transactions on Computer and Communication Systems*, 10(2), 39–46. Korea Information Processing Society.

[15] Denysiuk, D., Geidarova, O., Kapustian, M., Lysenko, S., & Sachenko, A. (2023). Blockchain-based deep learning algorithm for detecting malware. *IntelITSIS*, 529–538.

[16] Kumar, R., Wang, W., Kumar, J., Yang, T., & Ali, W. (2021). Collective intelligence: Decentralized learning for Android malware detection in IoT with blockchain. *arXiv preprint arXiv:2102.13376*.

[17] Martin, G., Spencer, D., Hair, A., K, D., Laudanna, S., P, V., & Visaggio, C. A. (2022). Mobile malware detection using consortium blockchain. *Artificial Intelligence for Cybersecurity*, 137–160. Springer.

[18] Gu, J., Sun, B., Du, X., Wang, J., Zhuang, Y., & Wang, Z. (2018). Consortium blockchain-based malware detection in mobile devices. *IEEE Access*, 6, 12118–12128. IEEE.

[19] Sheela, S., Shalini, S., Harsha, D., Chandrashekar, V. T., & Goyal, A. (2023). Decentralized malware attacks detection using blockchain. *ITM Web of Conferences*, 53, 03002. EDP Sciences.

[20] Cui, Y., Sun, Y., Lin, Z., Ma, B., & Li, Y. (2023). Potentially unwanted app detection for blockchain-based Android app marketplace. *IEEE Internet of Things Journal*, 10(24), 21154–21167. IEEE.

[21] Gupta, S., Thakur, P., Biswas, K., Kumar, S., & Singh, A. P. (2021). Developing a blockchain-based and distributed database-oriented multi-malware detection engine. *Machine Intelligence and Big Data Analytics for Cybersecurity Applications*, 249–275. Springer.

[22] Wressnegger, C., Freeman, K., Yamaguchi, F., & Rieck, K. (2017). Automatically inferring malware signatures for anti-virus assisted attacks. *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 587–598.

[23] Senanayake, J., Kalutarage, H., Petrovski, A., Piras, L., & Al-Kadri, M. O. (2024). Defendroid: Real-time Android code vulnerability detection via blockchain federated neural network with XAI. *Journal of Information Security and Applications*, 82, 103741. Elsevier.

[24] Park, J. H., Singh, S. K., Salim, M. M., Azzaoui, A. E., & Park, J. H. (2022). Ransomware-based cyber attacks: A comprehensive survey. *Journal of Internet Technology*, 23(7), 1557–1564.

[25] Kalphana, K. R., Aanjankumar, S., Surya, M., Ramadevi, M. S., Ramela, K. R., Anitha, T., Nagaprasad, N., & Krishnaraj, R. (2024). Prediction of android ransomware with deep learning model using hybrid cryptography. *Scientific Reports*, 14(1), 22351. Nature Publishing Group UK London.

[26] Hyperledger. (2024). Hyperledger Fabric. Retrieved from [https://github.com/hyperledger/fabric] (Accessed: Sep 10, 2024).

[27] Cisco Talos. (2024). ClamAV. Retrieved from [https://github.com/Cisco-Talos/clamav] (Accessed: Sep 10, 2024).

[28] Bhatia, A. (2020). Collection of android malware samples. Retrieved from [https://github.com/ashishb/android-malware] (Accessed: Oct 20, 2024).

[29] Divested Computing Group. (2024). Hypatia. Retrieved from [https://f-droid.org/en/packages/us.spotco.malwarescanner/] (Accessed: Sep 10, 2024).

[30] Tuan, H. M., Hai, T. H. ., & Thu, P. H. (2023). A new study for global dynamics and numerical simulation of a discrete-time computer virus propagation model. *Journal of Science and Technology on Information Security*, *3*(20), 35-42. https://doi.org/10.54654/isj.v3i20.982

[31] Toan, N. N. ., Dung, L. T., & Thang, D. Q. (2022). Static Feature Selection for IoT Malware Detection. *Journal of Science and Technology on Information Security*, *1*(15), 74-84. https://doi.org/10.54654/isj.v1i15.844.

ABOUT THE AUTHORS

Nguyen Tan Cam

Workplace: University of Information Technology, Vietnam National University, Ho Chi Minh City, Vietnam

Email: camnt@uit.edu.vn

Education: He received the bachelor's degree in information technology from the University of Sciences, Vietnam National University Ho Chi Minh City in 2006. He received his master's degree in information systems from this university in 2010. He received his PhD degree of Information Technology from University of Information Technology, Vietnam National University Ho Chi Minh City (UIT VNU-HCM) in 2021. He is a lecturer at UIT VNU-HCM.

Recent research direction: mobile security, distributed network security, IoT security, and machine learning-based cyber security.

Tên tác giả: **Nguyễn Tấn Cầm**

Cơ quan công tác: (1) Trường Đại học Công nghệ thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh; (2) Đại học Quốc gia Thành phố Hồ Chí Minh.

Email: camnt@uit.edu.vn

Quá trình đào tạo: Tốt nghiệp Cử nhân Công nghệ thông tin tại Trường Đại học Khoa học tự nhiên, Đại học Quốc gia Thành phố Hồ Chí Minh năm 2006. Tốt nghiệp Thạc sĩ Hệ thống thông tin tại Trường Đại học Khoa học tự nhiên, Đại học Quốc gia Thành phố Hồ Chí Minh năm 2010. Tốt nghiệp Tiến sĩ Công nghệ thông tin tại Trường Đại học Công nghệ thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh năm 2021. Ông là giảng viên của Trường Đại học Công nghệ thông tin, Đại học Quốc gia TP.HCM.

Hướng nghiên cứu hiện nay: bảo mật di động, bảo mật mạng, học máy và học sâu.

Pham Nhat Duy

Workplace: University of Information Technology, Vietnam National University, Ho Chi Minh City, Vietnam

Email: duypn@uit.edu.vn

Education: Received bachelor degree in 2021 and Master of Information Technology in 2024.

Recent research direction: android security, malware detection and web security.

Tên tác giả: **Phạm Nhật Duy**

Cơ quan công tác: (1) Trường Đại học Công nghệ thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh; (2) Đại học Quốc gia Thành phố Hồ Chí Minh.

Email: duypn@uit.edu.vn

Quá trình đào tạo: Tốt nghiệp cử nhân ngành Công nghệ thông tin năm 2021, thạc sĩ ngành Công nghệ thông tin năm 2024 tại Trường Đại học Công nghệ thông tin, Đại học Quốc gia TP.HCM.

Hướng nghiên cứu hiện nay: bảo mật Android, phát hiện mã độc và bảo mật web.

Hoang Mai Thien Phuc

Workplace: University of Information Technology, Vietnam National University, Ho Chi Minh City, Vietnam

Email: 20520695@gm.uit.edu.vn

Education: Received Bachelor of Information Technology in 2024.

Recent research direction: android security, malware detection and web security

Tên tác giả: **Hoàng Mai Thiên Phúc**

Cơ quan công tác: (1) Trường Đại học Công nghệ thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh; (2) Đại học Quốc gia Thành phố Hồ Chí Minh.

Email: 20520695@gm.uit.edu.vn

Quá trình đào tạo: Tốt nghiệp cử nhân ngành Công nghệ thông tin năm 2024 tại Trường Đại học Công nghệ thông tin, Đại học Quốc gia TP.HCM.

Hướng nghiên cứu hiện nay: bảo mật Android, phát hiện mã độc và bảo mật web.