Article

Not peer-reviewed version

Using ASCON-Based Fuzzy Hashing for Efficient Malware Analysis

Noman Minhas * and Maida Naveed

Posted Date: 6 August 2024

doi: 10.20944/preprints202408.0263.v1

Keywords: Cryptography; Malware Analysis; ASCON

Article

Using ASCON-Based fuzzy Hashing for Efficient Malware Analysis

Noman Nasir Minhas * and Maida Naveed

Department of Cyber Security, Air University, Islamabad, Islamabad, Pakistan; maida.naveed@au.edu.pk

* Correspondence: contact.nomanminhas@gmail.com

Abstract: The rapid proliferation of digital devices and connectivity has seen an unprecedented rise in the threats of cyber-attacks, making cybersecurity a high research priority worldwide. Among the various forms of cyberattacks, malicious software or malware, significantly disrupts the privacy and integrity of data on millions of computing devices globally. It is already a well-established notion that robust malware analysis underpins successful protection systems in the rapidly changing threat landscape. Traditional malware detection techniques, although effective against known malware types, are often found wanting against new and sophisticated varieties. The situation necessitates the exploration of advanced techniques like cryptographic hashing and fuzzy hashing that have proven instrumental in handling malware's dynamic nature. Fuzzy hashing, by allowing a score-based match rather than an exact digital signature, enables the identification of unknown malware that are derivatives of known malwares. However, the efficiency of the fuzzy hashing algorithms comes to the forefront concerning the encryption used. The research paper titled, "Using ASCON-based Fuzzy Hashing for Efficient Malware Analysis," seeks to explore leveraging the ASCON encryption, a lightweight but secure encryption standard, in the application of fuzzy hashing for malware detection.

Keywords: cryptography; malware analysis; ASCON

---

1. Introduction

The rapid proliferation of digital devices and connectivity has seen an unprecedented rise in the threats of cyber-attacks, making cybersecurity a high research priority worldwide. Among the various forms of cyberattacks, malicious software or malware, significantly disrupts the privacy and integrity of data on millions of computing devices globally. It is already a well-established notion that robust malware analysis underpins successful protection systems in the rapidly changing threat landscape.

Traditional malware detection techniques, although effective against known malware types, are often found wanting against new and sophisticated varieties. The situation necessitates the exploration of advanced techniques like cryptographic hashing and fuzzy hashing that have proven instrumental in handling malware's dynamic nature. Fuzzy hashing, by allowing a score-based match rather than an exact digital signature, enables the identification of unknown malware that are derivatives of known malwares. However, the efficiency of the fuzzy hashing algorithms comes to the forefront concerning the encryption used.

The research paper titled, "Using ASCON-based Fuzzy Hashing for Efficient Malware Analysis," seeks to explore leveraging the ASCON encryption, a lightweight but secure encryption standard, in the application of fuzzy hashing for malware detection.

2. Research Motivation

The use of advanced cryptographic algorithms for secure communications and data storage has been widely researched and documented. Among them, ASCON has emerged as a promising alternative owing to its high security and performance, especially in constrained environments. The combination of this encryption technique with fuzzy hashing for the purpose of malware identification and analysis is an unchartered research area, which forms the crux of this study.

 doi:10.20944/preprints202408.0263.v1

Moreover, literature has a sufficient number of works employing other cryptographic algorithms for fuzzy hashing, but the inclusion of ASCON in scenario remains scantily explored. Thus providing both an opportunity and a necessity for an in-depth study.

The motivation behind the research is threefold. Firstly, with malware becoming more sophisticated and diverse, there is a constant need to upgrade and refine detection and analysis techniques such as fuzzy hashing. Secondly, the potential advantages of using a lightweight algorithm like ASCON in constrained systems represent an exciting avenue to explore. Finally, research in this area could significantly contribute to the further strengthening of cybersecurity systems, fulfilling the increasing need for more robust safeguards against cyber threats.

3. Literature Review

Li, Y., Sundaramurthy, S.C., Bardas, A.G., Ou, X., Caragea, D., Hu, X. and Jang, J., 2015 [1] the paper examines fuzzy hashing algorithms for clustering malware and identifies limitations in current methods. It proposes a new fuzzy hash function inspired by current algorithms and advanced malware clustering approaches, which outperforms existing ones. The study also compares the performance of nextGen-hash and nextGen-hash-imp algorithms, extracting different features from malware samples. The results show that "import table entries" features are more representative and efficient than low-level n-gram features in comparing overall similarity, suggesting a potential improvement in malware clustering.

Rodriguez-Bazan, H., Sidorov, G. and Escamilla-Ambrosio, P.J., 2023 [2]. The research presents a new approach for classifying Android ransomware using an image-based Convolutional Neural Network (CNN). Using Natural Language Processing (NLP) and Fuzzy Hashing, the method converts an Android Application Package (APK) into a grayscale image. When tested on a dataset of 7,765 ransomware samples, the technique outperformed previous methods in the literature.

Shiel, I. and O'Shaughnessy, S., 2019 [3]. In order to get around the drawbacks of file-level similarity hashing, the study investigates the use of section-level similarity hashing to categorize malware variations. Studies carried out on well-known malware families demonstrate that, when it comes to identifying malware in Windows Portable Executable (PE) files, section-level hashing and comparison perform noticeably better than file-level hashing.

Tariq, U., Ahmed, I., Khan, M.A. and Bashir, A.K., 2023 [4]. The paper covers the difficulties and weaknesses in Internet of Things security and suggests workable solutions, such as blockchain technology and quantum cryptography in addition to more established encryption techniques. By classifying attacks and vulnerabilities, examining attack techniques, and presenting security solutions through case studies of important IoT applications, it seeks to serve as an invaluable resource for researchers.

Aghili, S.F., Sedaghat, M., Singelée, D. and Gupta, M., 2022 [5]. This paper presents the method and its limitations for classifying malware variations in Windows Portable Executable (PE) files. It does not evaluate how well it works in contrast to malware that has been disguising itself, which may be difficult to find using standard methods. There might be an extra computational expense because the method splits malware executables into binary headers and sections. It also has limited generality to unknown malware families because it applies a database of existing malware families to classify new samples. For the process to be successful, section information must be present in the malware executables, which may be deliberately removed or obfuscated to evade detection.

Naik, N., Jenkins, P. and Savage, N., 2019 [6]. This study presents a method for detecting new or undiscovered ransomware variants using fuzzy hashing. The method uses three fuzzy hashing methods (SSDEEP, SDHASH, and mvHASH-B) on a WannaCry or WannaCryptor malware corpus. The success rate of similarity detection and fuzzy similarity scores are used to assess the effectiveness of the approaches. The results are compared to determine the relative accuracy of the chosen fuzzy hashing algorithm. This method is crucial for organizations to focus on ransomware prevention, as it denies access to data and poses a significant threat to their information systems. The results are used

 doi:10.20944/preprints202408.0263.v1

to determine the relative accuracy of the chosen fuzzy hashing algorithm alongside their malware prevention strategy.

Naik, N., Jenkins, P., Savage, N., Yang, L., Boongoen, T. and Iam-On, N., 2020, July [7]. The research offers a methodology called fuzzy-import hashing, which combines fuzzy hashing with import hashing approaches to detect and analyze malware. This integration seeks to increase detection accuracy while minimizing performance impact. Experiments on gathered malware and goodware corpus, comparative evaluation against YARA rules, and use in fuzzy c-means clustering demonstrate the suggested technique.

Naik, N., Jenkins, P., Savage, N., Yang, L., Boongoen, T., Iam-On, N., Naik, K. and Song, J., 2021 [8]. The research presents two ways for improving the efficiency of YARA rules for malware detection without increasing complexity and overheads: upgraded YARA rules and embedded YARA rules. These methods use fuzzy hashing and fuzzy rules to estimate the chance of a file containing malware, resulting in better detection results than standard YARA rules.

Naik, N., Jenkins, P., Savage, N., Yang, L., Naik, K., Song, J., Boongoen, T. and Iam-On, N., 2020, December [9]. The study proposes employing fuzzy hashing alongside basic YARA rules to improve the detection rate of YARA rules for malware triaging. By comparing the detection rate of upgraded YARA rules to existing triaging approaches such as fuzzy hashing and import hashing, the suggested strategy is proven to improve overall triaging results.

Chang, D., Hong, D., Kang, J. and Turan, M.S., 2022 [10]. The study investigates the Ascon family's resistance to conditional cube assaults in a nonce-misuse environment. It introduces novel state- and key-recovery attacks on the Ascon family, such as Ascon-128a and Ascon-80pq. The attacks recover the whole state and secret key of Ascon-128a in less cycles, exceeding the designers' data limit. Ascon-128's incomplete state information can also be restored. Furthermore, given that the whole state information of Ascon-80pq was obtained in a prior assault, the study demonstrates that the Ascon-80pq's 160-bit secret key can be recovered. These assaults shed light on Ascon's security in a non-misuse environment.

Rodriguez-Bazan, H., Sidorov, G. and Escamilla-Ambrosio, P.J., 2023 [2]. The research provides a new method for identifying Android ransomware based on photos using a Convolutional Neural Network (CNN) generated by converting an Android Application Package (APK) into a grayscale image using Natural Language Processing (NLP) techniques and Fuzzy Hashing. The suggested method was evaluated on a dataset of 7,765 Android ransomware samples and outperformed previous methods in the literature in terms of accuracy.

Mostafa, A.M., Ezz, M., Elbashir, M.K., Alruily, M., Hamouda, E., Alsarhani, M. and Said, W., 2023 [11]. The paper describes an adaptive multi-factor multi-layer authentication architecture for cloud user authentication, with the goal of improving cloud platform security and preventing unwanted access and data breaches. The framework includes access control, intrusion detection algorithms, and automatic authentication method selection. To improve identity verification, it employs several authentication elements such as user factors, geolocation, and browser confirmation. To protect login information, an AES-based encryption component is also used. The suggested approach performs admirably in identifying potentially harmful users and intruders, hence efficiently preventing intentional attacks on cloud services and data.

Rodriguez-Bazan, H., Sidorov, G. and Escamilla-Ambrosio, P.J., 2023 [2]. The study proposes a framework for safe authentication in cloud systems, however its assessment is purely theoretical, with no specific analysis of its performance or scalability. When compared to simpler techniques, the framework's multi-factor multi-layer authentication strategy may add complexity and difficulty for consumers. This limitation is not addressed in the study, nor are options for streamlining the authentication process without compromising security explored. The behavioral analysis component of the system is dependent on accurate profiling of user behavior patterns, which may be sensitive to changes in user behavior or discrepancies in data gathering. The framework could also be prone to social engineering attacks, which could be avoided by user education and awareness training. The

 doi:10.20944/preprints202408.0263.v1

article does not go into detail on the integration procedure or its interoperability with existing cloud systems.

Pagani, F., Dell'Amico, M. and Balzarotti, D., 2018, March [12]. The research compares four prominent fuzzy hashing algorithms in various contexts, finding that ssdeep outperforms competing methods. The optimal algorithm for identifying similarities across binary program files depends on the specific use case situation, as no study has determined the best suited algorithm for this purpose.

Namanya, A.P., Awan, I.U., Disso, J.P. and Younas, M., 2020 [13]. Because harmful files are evading existing security systems, this research investigates merging different hashing approaches to offer a quantifiable malware score and achieve higher detection rates. The proposed approach for malware rating based on hash results improves true detection rates of malware significantly (> 90%).

Ali, H., Batool, K., Yousaf, M., Islam Satti, M., Naseer, S., Zahid, S., Gardezi, A.A., Shafiq, M. and Choi, J.G., 2022 [14]. The study presents a method for identifying malicious Android applications utilizing repacked malicious code and fuzzy reasoning for categorization. The suggested methodology outperforms existing similar approaches with a detection rate of roughly 74% for repacked malware.

Samra, A.A.A., Qunoo, H.N., Al-Rubaie, F. and El-Talli, H., 2019, March [15]. The report includes a survey that compares the two leading static Android malware detection approaches: permission-based detection and signature-based detection. The study's goal is to give scholars a comprehensive knowledge of the parallels, differences, and correctness of key published research in this subject.

Aslan, Ö.A. and Samet, R., 2020 [16]. The study examines several malware detection tactics and procedures, highlighting their advantages and disadvantages. It underlines the difficulties in identifying both known and unknown malware, as well as the necessity for fresh research and methodologies in this field.

Amira, A., Derhab, A., Karbab, E.B. and Nouali, O., 2023 [17]. The study examines the use of community detection algorithms in malware analysis to detect malware families and variations rather than individual instances of malware, which can reduce detection time dramatically. The survey examines cutting-edge malware analysis solutions that use community detection algorithms and offers a taxonomy that categorizes the solutions based on analysis task, community detection approach, target platform, analysis kind, and source of features. It also indicates research gaps as well as potential future research directions.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 August 2024 doi:10.20944/preprints202408.0263.v1

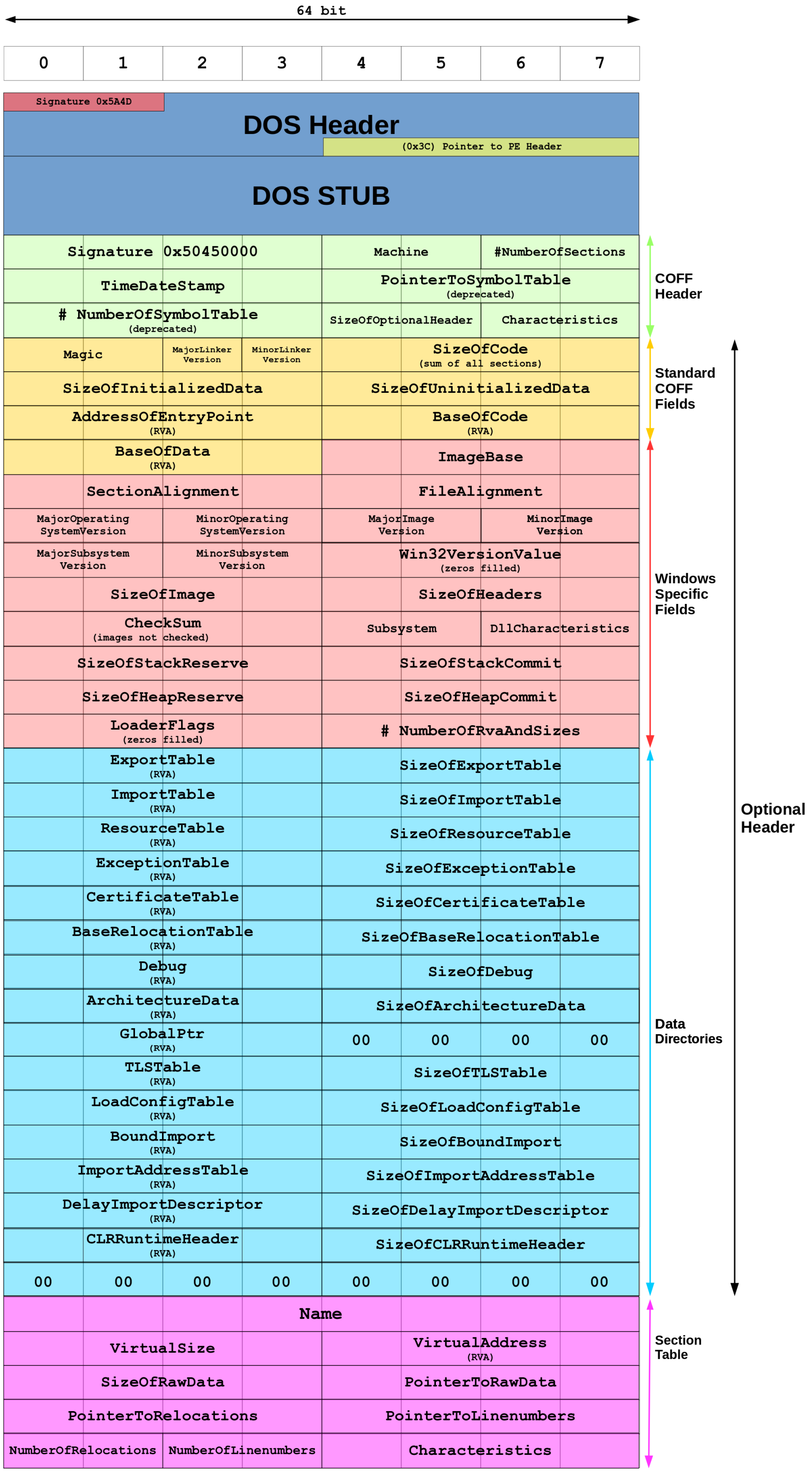

Figure 1. Structure of a PE File

 doi:10.20944/preprints202408.0263.v1

Paper Name	Method and Techniques	Contributions	Results	Primary Limitation
Experimental Study of Fuzzy Hashing in Malware Clustering Analysis	The paper examines fuzzy hashing techniques for malware similarity analysis, addressing issues and proposing new methods. It also explores block-based distance computation and uses nextGen-hash and nextGen-hash-imp algorithms to evaluate performance variations influenced by different features in a malware dataset.	The paper discusses the challenge of automating malware grouping using fuzzy hashing algorithms, highlighting the lack of rigorous experimentation in existing research. It examines existing fuzzy hashing algorithms and their limitations in malware clustering, proposing novel ways to construct these algorithms that outperform existing ones.	The paper critiques current fuzzy hashing algorithms for malware similarity analysis and suggests improved performance by considering input data structure, high-level semantic features, and comparing Bloom filters and feature hashing. It also introduces a new approach called feature hashing"."	The study evaluates popular fuzzy hashing algorithms but lacks a wider range of options, limiting generalization. It also doesn't consider malware families' impact on performance, making it unclear how the findings apply to different types of malware.
Android Ransomware Analysis Using Convolutional Neural Network and Fuzzy Hashing Features	The study presents a method for classifying malware using data pretreatment and natural language processing techniques. It converts Android APK files into grayscale photos, extracts and cleans content using Natural Language Processing algorithms, and trains a convolutional neural network to classify malware families.	A new method for categorizing Android ransomware uses Natural Language Processing (NLP) techniques, Fuzzy Hashing, and Convolutional Neural Network (CNN) for malware classification. Experimental testing on 7,765 ransomware samples demonstrates the method's accuracy, focusing on ransomware's potential maliciousness and making recovery difficult without payment.	The proposed method for Android ransomware categorization uses Convolutional Neural Network (CNN) and fuzzy hashing characteristics, achieving higher accuracy than existing methods. With 7,765 samples, the CNN model achieved an average accuracy of 98.16%, outperforming other machine learning techniques like k-NN, Random Forest, Multilayer Perceptron, and Support Vector Machine.	The paper uses a small dataset of 1,000 Android samples for ransomware detection, which may limit its applicability in real-world scenarios. The model's precision, recall, and F1-score metrics are not comprehensive, and its computationally expensive convolutional neural network may result in false positives, making it unsuitable for real-time scenarios.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 August 2024 doi:10.20944/preprints202408.0263.v1

Paper Name	Method and Techniques	Contributions	Results	Primary Limitation
Improving file-level fuzzy hashes for malware variant classification	The study identifies malware variations in Windows Portable Executable files by dividing malware into binary headers and sections, and applying a similarity digest to each part. Experiments show section-level hashing is the most effective method, but the F-measure measures effectiveness.	The paper introduces section-level similarity hashing as a technique for malware variant classification, addressing the limitations of file-level hashing. The study evaluates the effectiveness of similarity hashing at file and section levels, demonstrating its advantages and the viability of classifying malware variants using similarity digests.	The study used file and section level digests to predict malware family membership in Windows Portable Executable files. Results showed section-level hashing and comparison outperformed file-level hashing in malware classification, particularly recall. The non-obfuscated segment method performed better. The study demonstrated that section-level similarity hashing can overcome file-level hashing constraints and is feasible for malware classification.	The paper presents a method for categorizing malware variations in Windows Portable Executable files, but its effectiveness is limited due to its inability to identify obfuscated malware. Section-level fuzzy hashing may have higher computational overhead, and its generalization to unknown malware families is limited. Section information in malware executables may be intentionally erased or obfuscated.
Fortifying IoT against crimpling cyber-attacks: a systematic review	The paper uses a systematic approach to evaluate security challenges in IoT and create a comprehensive threat taxonomy. It conducts a thorough literature search using relevant keywords across reputable databases. The authors select key network security topics based on their expertise. The study evaluates the effectiveness of existing security features and protocols in mitigating IoT risks and proposes improvements to address identified gaps.	The article provides a comprehensive analysis of IoT security flaws and attacks, focusing on innovative technologies like blockchain and QC-PUFs. It reviews existing research, evaluates the effectiveness of security features and protocols, and suggests incorporating blockchain and machine learning algorithms to enhance IoT security.	The study analyzes IoT security research, identifying vulnerabilities and attacks based on their intended recipients. It provides a comprehensive analysis of attack techniques and suggests defenses. The study presents case studies of IoT applications, assesses security features, suggests improvements, and uses survey data to identify new security issues. It distinguishes itself from surveys by using an extensive research methodology.	This work presents a technique for categorizing malware variations in Windows Portable Executable files, but it has drawbacks. It doesn't compare it to disguised malware, divides malware executables into binary headers and sections, incurs computational costs, and has limited generalization to unknown malware families. The technique requires malware executables to contain section information and can only identify malware from a small dataset of 2,400 samples.

Paper Name	Method and Techniques	Contributions	Results	Primary Limitation
MLS-ABAC: Efficient Multi-Level Security Attribute-Based Access Control scheme	In order to comply with the requirements of the NIST Attribute-Based Access Control (ABAC) model, the paper presents the Multi-Level Security ABAC (MLS-ABAC) scheme. It presents both conceptual and formal models and employs an outsourceable Ciphertext-Policy ABE system for decryption. A constant ciphertext size and encryption and decryption times of 18 and 10 ms are examples of performance measures.	The study introduces a Multi-Level Security ABAC (MLS-ABAC) scheme that effectively controls access to sensitive data in an IoT context, adhering to NIST's Attribute-Based Access Control (ABAC) model standards. The approach includes both conceptual and formal models and uses an outsourceable Ciphertext-Policy ABE scheme for decryption. Experimental data shows an efficient encryption and decryption running time of 18-10 ms.	Experimental results show that the MLS-ABAC scheme fits the requirements of NIST's ABAC model for IoT application. It achieves a constant ciphertext size of 230 bytes and efficient encryption and decryption running times of 18 and 10 ms.	The proposed access control approach has several drawbacks, including its policy function being limited to AND-gate circuits, slow decryption technique, and lack of revocation challenge. The algorithm is inefficient and does not address user revocation, covert channel attacks, or sanitization. The technique also lacks clear methods for confirming encrypted data accuracy and does not address security risks associated with outsourcing decryption services to external cloud providers.
A Ransomware Detection Method Using Fuzzy Hashing for Mitigating the Risk of Occlusion of Information Systems	The paper uses fuzzy hashing for ransomware detection, using three methods: SSDEEP, SDHASH, and mvHASH-B. It examines the success rate of each method in detecting similarity and clusters the collected ransomware corpus. The findings are compared to determine the relative accuracy of the chosen fuzzy hashing methods, focusing on the WannaCry or WannaCryptor malware.	The study presents a fuzzy hashing-based ransomware detection strategy to mitigate the risk of information system attacks. It considers the polymorphic behavior and dispersion of ransomware variants, utilizing three algorithms (SSDEEP, SDHASH, and mvHASH-B) to assess similarity identification success rates and cluster the collected ransomware corpus.	The study presents a ransomware detection method using fuzzy hashing methods SSDEEP, SDHASH, and mvHASH-B to assess similarity identification success rates and cluster a ransomware corpus. This method aims to reduce information system blockage caused by ransomware attacks by comparing new or unknown variants with existing samples, providing insight into the chosen fuzzy hashing algorithms' accuracy.	The proposed solution detects ransomware at the file level, but may not be sufficient for all attacks. Ransomware often uses tactics to avoid detection, like inserting malicious code or modifying its structure. The study focuses on recognizing known ransomware variations and does not discuss fuzzy hashing's effectiveness against polymorphic ransomware. Fuzzy hashing's effectiveness depends on target file availability and may not scale effectively for large data amounts.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 August 2024 doi:10.20944/preprints202408.0263.v1

Paper Name	Method and Techniques	Contributions	Results	Primary Limitation
Fuzzy-Import Hashing: A Malware Analysis Approach	The research introduces fuzzy-import hashing, a combination of fuzzy and import hashing algorithms for malware identification and analysis. Studies, including malware and goodware corpus examination, YARA rule comparison, and fuzzy c-means clustering implementation, demonstrate its success.	The research proposes a fuzzy-import hashing strategy for malware identification and analysis, combining fuzzy hashing with import hashing methods to improve detection accuracy without affecting overall analysis performance. This strategy offers higher detection rates and generates fuzzified data for later clustering or classification. Studies show the efficacy of this method, including malware and goodware corpus examination.	The fuzzy-import hashing strategy, which combined fuzzy and import hashing, improved malware detection rates. This technique was tested on a malware and goodware corpus, and its efficacy was confirmed through experiments, comparative evaluation against YARA rules, and fuzzy c-means clustering.	Fuzzy-import hashing, a method for identifying malware, may be ineffective against disguised malware and prone to false positives. Its success depends on the availability and correctness of import table information, which can be computationally expensive in real-time virus detection applications. The study focuses on detecting known malware variants, but a more comprehensive evaluation would include testing against a broader range of malware samples, including polymorphic and extensively obfuscated version

 doi:10.20944/preprints202408.0263.v1

Paper Name	Method and Techniques	Contributions	Results	Primary Limitation
Embedded YARA rules: strengthening YARA rules utilising fuzzy hashing and fuzzy rules for malware analysis	The study suggests two ways to enhance the effectiveness of YARA rules in malware detection: upgraded YARA rules and embedded YARA rules. Upgraded YARA rules use fuzzy hashing to determine file malware risk, while integrated YARA rules combine fuzzy hashing with fuzzy rules, considering imprecise or ambiguous data. The expanded YARA rules technique uses the Damerau-Levenshtein distance measure.	The study proposes two strategies to enhance the performance and detection of YARA rules in malware analysis: enhanced YARA rules and embedded YARA rules. The enhanced method uses fuzzy hashing to evaluate files not recognized as malware, while the embedded technique uses fuzzy hashing and fuzzy rules to produce probabilistic results. The experimental results show these strategies are more effective than simple YARA guidelines.	The enhanced YARA rules based on SSDEEP showed slightly better total similarity detection rate (67.1 compared to standard YARA rules, 62.2), but the improvement was marginal. The embedded YARA rules, which mix fuzzy hashing and fuzzy rules, performed marginally better or worse. The optimum analysis method for all cases is challenging, indicating further research is needed.	The proposed embedded YARA rules approach relies on fuzzy rule correctness, which is determined by the quality and completeness of the underlying fuzzy rule base. However, its effectiveness may be reduced if fuzzy rules are not well-defined or do not capture malware features. The approach's generalization to unknown malware families is limited, potentially leading to false negatives. Additionally, processing overhead can affect real-time malware analysis capabilities when dealing with large files.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 August 2024 doi:10.20944/preprints202408.0263.v1

Paper Name	Method and Techniques	Contributions	Results	Primary Limitation
Fuzzy Hashing Aided Enhanced YARA Rules for Malware Triaging	The paper uses three triaging methods to assess malware samples: fuzzy hashing, import hashing, and YARA rules. Fuzzy hashing produces fuzzy hash values for ransomware samples, comparing them to previously discovered samples. Import hashing generates IMPHASH hash values, comparing them to previously detected samples. YARA rules use pattern matching to detect and classify malware, with an improved version adding fuzzy hashing to improve detection rates.	The research presents fuzzy hashing-aided YARA rules to enhance detection rates without complexity or overheads. It analyzes three triaging approaches on ransomware samples: fuzzy hashing, import hashing, and YARA rules. When one method fails, the methodology supplements basic YARA rules, increasing detection rates. The article emphasizes the need for efficient, resource-optimized methodologies for malware analysis.	The study compares the detection rate of enhanced YARA rules to various triaging approaches, finding that enhanced YARA rules marginally outperform basic YARA rules (67.1%). The study highlights the need for efficient and resource-optimized methodologies for analyzing large malware volumes. The study applies three triaging approaches: fuzzy hashing, import hashing, and YARA rules, focusing on malware samples for extended experimentation.	The proposed YARA rules using fuzzy hashing may not be effective in detecting extensively obfuscated malware due to changes in file structure and signatures. Fuzzy hashing is prone to false positives and may mistake innocent files for malware if they share structural similarities with known samples. The success of the method depends on the chosen parameters, such as similarity threshold and hashing algorithm. The method may also miss threats due to incorrect categorization of malware samples from new or unknown families.

 doi:10.20944/preprints202408.0263.v1

Paper Name	Method and Techniques	Contributions	Results	Primary Limitation
Resistance of Ascon Family Against Conditional Cube Attacks in Nonce-Misuse Setting	The paper explores the Ascon family's resistance to conditional cube attacks in a nonce-misuse context. It presents new state- and key-recovery attacks on the family, including Ascon-128a and Ascon-80pq. These attacks recover the complete state and secret key of Ascon-128a in less cycles, exceeding the designers' data limit. The study also demonstrates the recovery of partial state information and the 160-bit secret key, revealing Ascon's security in a nonce-misuse scenario.	The study presents new state- and key-recovery attacks against the Ascon family, including Ascon-128a and Ascon-80pq, in a nonce-misuse scenario. These attacks recover the complete state and secret key of Ascon-128a in less cycles, exceeding the designers' data limit. The authors also demonstrate the recovery of partial state information and the 160-bit secret key, provided the complete state information was acquired in a previous attack. These attacks provide valuable insights into the Ascon family's security vulnerabilities and weaknesses.	In a nonce-misuse situation, the study proposes new state- and key-recovery attacks against the Ascon family, including Ascon-128a and Ascon-80pq. These attacks recover the complete state and secret key of Ascon-128a in less cycles, exceeding the designers' data limit. They also demonstrate the recovery of Ascon-128 partial state information and the Ascon-80pq 160-bit secret key, provided complete state information was acquired in a previous attack. These assaults provide useful insights into the Ascon family's security and emphasize the need for additional investigation and enhancements.	The study evaluates the Ascon family's resistance to conditional cube attacks, but doesn't cover its overall security against other attacks. It mainly focuses on 3-round and 4-round examples, but could be improved by investigating higher-round instances. The research assumes perfect data alignment and sufficient memory, which may not be practical in real-world situations. Comparing conditional cube attacks to real-world implementations would provide more insights.

 doi:10.20944/preprints202408.0263.v1

Paper Name	Method and Techniques	Contributions	Results	Primary Limitation
Android Ransomware Analysis Using Convolutional Neural Network and Fuzzy Hashing Features	A new method for categorizing Android ransomware uses Natural Language Processing (NLP) techniques and Fuzzy Hashing, followed by a Convolutional Neural Network (CNN) for malware classification. The method outperforms other methods in accuracy, as demonstrated in experimental testing on 7,765 Android ransomware samples. The research focuses on ransomware, encrypting files with unknown keys, making recovery difficult without payment.	The study presents a new method for identifying Android ransomware using Convolutional Neural Network (CNN) and images generated from an Android Application Package (APK) using Natural Language Processing (NLP) techniques and Fuzzy Hashing. The method outperforms other methods in accuracy, as demonstrated in experimental testing on 7,765 ransomware samples. The study also highlights the use of NLP for text cleaning and extraction, enhancing accuracy.	The Convolutional Neural Network (CNN) method has been proven to be more accurate than existing methods in categorizing Android ransomware on a dataset of 7,765 samples. The CNN model achieved 94.37% accuracy when using the complete dataset and 98.16% when using the five most representative classes. It outperformed other machine learning models and state-of-the-art studies in Android malware analysis.	The study presents a methodology for detecting ransomware using 1,000 Android samples, but it lacks evaluation and is based on fuzzy hashing features, which can lead to false positives. The model is computationally expensive and unsuitable for real-time ransomware detection due to its reliance on a known malware database. The efficiency of section-level fuzzy hashing depends on the availability of section information in malware executables.

Paper Name	Method and Techniques	Contributions	Results	Primary Limitation
Beyond Precision and Recall: Understanding Uses (and Misuses) of Similarity Hashes in Binary Analysis	The study compares four popular fuzzy hashing methods in various scenarios, including shared source code and statically-compiled files. It examines low-level compilation process elements and technicalities to explain variances. The widely used ssdeep algorithm is found to perform poorly, and the optimal method is determined by the individual use case scenario.	The study examines fuzzy hash families in binary analysis case studies, focusing on identifying libraries, software, and detecting programs. It aims to explain individual findings and present examples of each algorithm's success or failure. The study addresses past work's weaknesses by highlighting unexpected results and performance variances, emphasizing that the best algorithm depends on the use case situation.	The study compares four fuzzy hashing methods in various contexts, including comparing programs with a large proportion of source code, connecting statically-compiled files with libraries, compiling files with different flags or compilers, and comparing programs with a high part of source code. The ssdeep method outperforms alternative algorithms, but no research definitively determines the best fuzzy hashing algorithm for program similarities, with the optimum choice varying based on the use case scenario.	The paper primarily discusses the use of similarity hashes in binary analysis, ignoring other techniques like signature-based analysis and machine learning. It highlights the lack of standardization and the potential for inconsistencies and misuse. The efficiency of similarity hashes depends on the hash settings, and the report doesn't provide detailed guidance on selecting appropriate parameters. The article also suggests integrating contextual information to improve the accuracy and relevance of results.

Paper Name	Method and Techniques	Contributions	Results	Primary Limitation
Similarity hash based scoring of portable executable files for efficient malware detection in IoT	The study investigates four hash types that are currently utilized in malware research for portable executable (PE) files.It is built and developed the recommended approach for malware scoring based on hash results. A number of experiments are conducted to determine the effectiveness of the proposed technique.	The research explores the use of multiple hashing techniques to generate a quantitative malware score and enhance detection rates. It addresses the issue of high false detection rates when using hashing techniques independently. The proposed approach, tested in experiments, results in a significant improvement (>90%) in malware detection rates.	The proposed approach for malware scoring based on hash results improved accurate detection rates of malware by more than 90%.	The paper describes a method for identifying malware variants in Windows Portable Executable (PE) files, which may not be applicable to other file formats such as ELF or Mach-O. The method's efficiency is not tested against obfuscated malware, which can change the structure of the virus. Section-level fuzzy hashing, which divides malware executables into binary headers and sections, might increase computational overhead, which may be an issue for real-time malware detection applications. Because it relies on a database of known malware families to categorize fresh samples, the method's generalization to unknown malware families is limited. The availability of section information in malware executables, which may be purposely erased or obfuscated to avoid detection, determines the usefulness of section information.

 doi:10.20944/preprints202408.0263.v1

Paper Name	Method and Techniques	Contributions	Results	Primary Limitation
Security Hardened and Privacy Preserved Android Malware Detection Using Fuzzy Hash of Reverse Engineered Source Code	The methodology involves identifying harmful Android app behavior, extracting package names from source code, matching them with known malware names, and computing fuzzy hashes for matched packages. It uses reverse engineering tools like dex2jar and jadx, extracting sensitive Java code features, and detecting malware using fuzzy hashes. Static analysis is used for inspection.	The study presents a method for detecting malicious Android applications using static analysis with fuzzy hashes and repacked dangerous code. The framework establishes a threshold value for malware detection, with a match larger than 40% indicating maliciousness. The method outperforms other methods with a detection rate of 74%.	The suggested approach identified 56 applications as malicious, 21 as suspicious, and 23 as benign. The studies were carried out on a dataset that included 3490 malware samples from 21 different families. When compared to other similar methodologies, the framework showed around 74% of the repacked malware.	The proposed Android malware detection method is based on the source code of the application, potentially limiting its application to closed-source ones. Fuzzy hashing, a technology known for producing false positives, may misclassify innocent apps as malware. The method's generalization to unknown malware families is limited, and code obfuscation techniques may avoid detection. The evaluation focuses on detecting known malware versions without substantial obfuscation and its effectiveness against heavily obfuscated malware.

Paper Name	Method and Techniques	Contributions	Results	Primary Limitation
Android Malware Classification Based on Fuzzy Hashing Visualization	The article uses a convolutional neural network (CNN) to categorize malware using grayscale images from the Android Application Package (APK). Natural language processing techniques are used for text cleaning, extraction, and fuzzy hashing. The decompiled smali code and manifest file are hashed using fuzzy hashing, and the CNN model is trained on grayscale images.	The study introduces a new method for classifying Android malware by converting APK samples into grayscale photos using natural language processing techniques. This method uses text cleaning, extraction, and fuzzy hashing to represent decompiled code as a set of hashes. The method yields higher accuracy (up to 98.24%) and was tested on an Android malware dataset containing 15,493 samples of five different malware types.	The proposed method for Android malware classification achieved 98.24% accuracy, using a fuzzy hashing approach and convolutional neural network model. This method reduced time-consuming operations and outperformed existing methods. Experiments were conducted on five malware varieties and their top classes, accounting for 81.12% of the dataset.	The research presents a method for categorizing malware samples using a small dataset of 2,400 Android samples. The model's accuracy is assessed using precision, recall, and F1-score, but its performance is not comparable to other Android malware classification methods. The model's fuzzy hashing characteristics can lead to false negatives, and its categorization findings are based on visual interpretation of fuzzy hashing visualizations, which may include subjectivity and errors. The model's generalization to unknown malware families is limited due to its focus on visualization and not subjectivity.

 doi:10.20944/preprints202408.0263.v1

Paper Name	Method and Techniques	Contributions	Results	Primary Limitation
A survey of Static Android Malware Detection Techniques	This study compares permission-based static Android malware detection strategies with signature-based techniques, focusing on similarities, differences, and accuracy. It evaluates machine learning technologies for high accuracy rates and various signature-based methodologies based on factors like tools, techniques, publishing year, and medium. The report provides an in-depth analysis of static Android malware detection strategies and a survey for researchers.	The study compares permission-based and signature-based static Android malware detection methods, highlighting their advantages and disadvantages. It emphasizes the need for further research to enhance malware detection results. The paper also discusses machine learning techniques and ROC curve evaluation for high accuracy in detecting malicious Android applications, aiming to provide researchers with a comprehensive understanding of static Android malware detection tools.	The article compares permission-based and signature-based malware detection methods for Android, highlighting their advantages. Signature-based detection has a higher True Positive Ratio (TPR) than permission-based detection, resulting in fewer undetected malwares. The research also highlights the use of machine learning techniques like Random Forest classifier for enhanced malware detection accuracy.	The document A Survey of Static Android Malware Detection Techniques" has limitations

 doi:10.20944/preprints202408.0263.v1

Paper Name	Method and Techniques	Contributions	Results	Primary Limitation
A Comprehensive Review on Malware Detection Approaches	The study explores malware detection tactics and new methods, including behavioral detection, data mining, stack computation tree predicate logic (SCTPL), model-checking algorithms, and heuristic-based detection algorithms. It provides insights through literature research, algorithm creation, and experimental results analysis, highlighting the effectiveness of these methods in detecting malware.	The study explores emerging technology developments in malware production and detection, examining various methods, their benefits and drawbacks, and the likelihood of identifying malware. It explores behavioral detection systems, algorithms, and heuristic-based detection systems. The goal is to help researchers understand malware detection technologies better and assess their efficiency.	The study explores various malware detection methods, highlighting the challenges in detecting complex malware and the need for new approaches. It introduces the stack computation tree predicate logic (SCTPL) for characterizing harmful behaviors and provides a model-checking algorithm for pushdown systems. The method is evaluated for accuracy and efficiency in identifying Android malware families. Further analysis is recommended for more reliable results and improved classification.	The research explores malware detection methods, highlighting the challenges in identifying complex malware and the need for innovative approaches. It introduces SCTPL for describing dangerous behaviors and presents a pushdown system model-checking algorithm, recommending further investigation for improved classification.

doi:10.20944/preprints202408.0263.v1

Paper Name	Method and Techniques	Contributions	Results	Primary Limitation
A Survey of Malware Analysis Using Community Detection Algorithm	The paper conducted a survey using manual keyword searches on popular search engines like IEEE, ACM, Scopus, and Google Scholar to identify important publications on malware, community identification, and cyberthreat. The articles were categorized into malware analysis, community detection, and malware analysis using algorithms. Unsupervised techniques were also used in community discovery approaches.	The research presents a new taxonomy for malware analysis solutions using community detection algorithms, comparing previous studies and revealing variations. It examines various malware analysis solutions, their benefits and drawbacks, classification criteria, primary duties, and function. The study also identifies challenges and suggests future research topics in malware analysis using community detection algorithms.	The poll indicates potential for advancement in malware analysis using community detection algorithms and addressing research gaps. The study provides taxonomy classification solutions for malware detection, classification, cyber-threat infrastructure detection, and feature selection for workstation and mobile platforms, addressing analysis tasks, approach, and feature source.	The study on community detection methods in malware research lacks a comprehensive overview of all methodologies and empirical examination, making it difficult to assess their practical usefulness. Additionally, it fails to address privacy and ethical issues associated with community detection algorithms, suggesting the need for further research to fully understand these strategies.}

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 August 2024 doi:10.20944/preprints202408.0263.v1

Paper Name	Method and Techniques	Contributions	Results	Primary Limitation
Strengthening Cloud Security: An Innovative Multi-Factor Multi-Layer Authentication Framework for Cloud User Authentication	The study presents an adaptive multi-factor multi-layer authentication framework for cloud user authentication, incorporating access control, intrusion detection systems, and automated method selection. It enhances identity verification using user factors, geolocation, and browser confirmation. An AES-based encryption component protects login information. The framework measures performance using false-positive and false-negative rates, reducing cloud-related risks like data leakage, brute force assaults, complexity, security, privacy, and execution time.	The paper presents an adaptive multi-factor multi-layer authentication architecture for cloud user authentication, aimed at enhancing security and preventing data breaches. The framework uses various authentication factors, geolocation, browser confirmation, and AES-based encryption to detect malicious users and intruders, preventing deliberate attacks on cloud services and data.	The study evaluates the effectiveness of an MFA framework and algorithm in detecting suspicious users and preventing malicious attacks on cloud servers or services. The results show low false-negative rates for varied user numbers and efficient detection of potentially harmful users and intruders. The study emphasizes the importance of authentication techniques in detecting legitimate users and preventing intrusions, thus enhancing the overall security of cloud platforms.	The study presents a framework for secure cloud system authentication, but its assessment is theoretical and lacks specific performance or scalability analysis. The multi-factor multi-layer authentication strategy may add complexity to consumers, and the study does not explore options for streamlining the process without compromising security. The behavioral analysis component relies on user behavior profiling, which may be sensitive to changes in user behavior or data gathering discrepancies.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 August 2024 doi:10.20944/preprints202408.0263.v1

4. Proposed Framework

The identification of malware variants and linking them to their respective Advanced Persistent Threat (APT) campaigns has proved to be a difficult problem that requires a sophisticated solution. This research proposes an efficient solution that leverages the capabilities of ASCON, a lightweight and robust cryptographic algorithm, along with fuzzy hashing techniques to analyze the structure and behavior of malware. The proposed solution targets to efficiently dissect and categorize malware through calculated hashes of various sections of a PE (Portable Executable) sample.

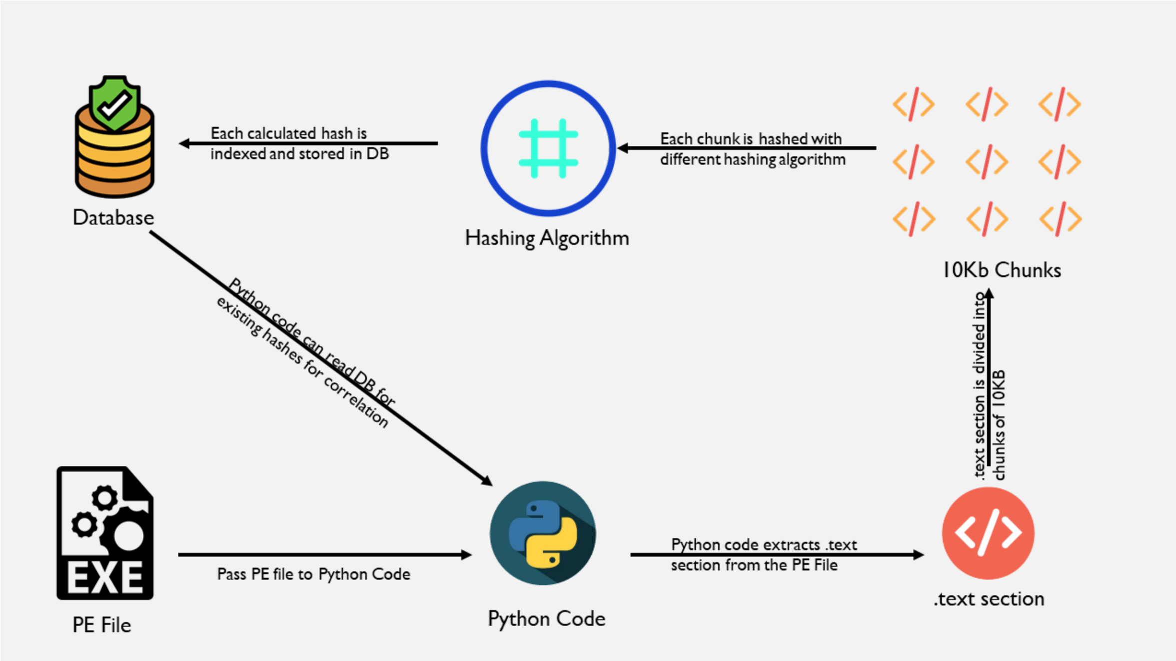

Figure 2. Proposed Framework

The proposed system integrates fuzzy hashing, an advanced technique that computes the cryptographic hash value for digital data, and produces a digital fingerprint which efficiently identifies similar but not identical data items. We propose to use this technique on PE samples, computing hash values for their different parts to allow identification and linking of different malware versions.

ASCON is a cryptographic algorithm that proves to be lightweight, highly secure, and notably, efficient in constrained environments. Thus, it emerges as an ideal candidate for this proposed system. ASCON-hashed PE segments can provide unique and versatile identifiers to link them with specific APT campaigns or identify variant versions.

The proposed system requires the division of the PE sample into distinct parts or sections. These divisions may be based on the functional or structural characteristics of the malware. Each portion is then hashed using the ASCON cryptographic algorithm separately rather than hashing the entire PE sample. This approach not only aids in identifying slight variations within a sample but also improves the efficiency of the analysis process.

By properly leveraging the proposed hashing method, similarities and linkages between different malware samples can be more effectively identified. The matching of such hashes can allow detection of reused components, variant versions, or even derivation patterns between different APT campaigns. Analysis of these patterns and components can then assist in tracing the origin of the malware or providing insights into the possible directions for the development and evolution of these threats.

The proposed solution's practicality and effectiveness are anticipated to be high, given the careful integration of fuzzy hashing and ASCON capabilities. The approach could significantly empower malware analysis, not only providing a more efficient method to handle a large volume of data but also offering a detailed dissection of malware structures and behaviors. This research provides a

 doi:10.20944/preprints202408.0263.v1

robust foundation for further investigation into the use of such methods in the rapid identification and classification of malware threats within increasingly complex cybersecurity landscapes.

5. System Algorithm

5.1. Function Analyze_PE_file(file_path: str)

- Read the PE file specified by file_path.
- Extract the .text section of the PE file.
- Divide the .text section into chunks of 10KB.

For each chunk:

- Calculate SHA-256 hash of the chunk.
- Calculate SHA-3 hash of the chunk.
- Calculate BLAKE2s hash of the chunk.
- Calculate Skein hash of the chunk.
- Calculate MD5 hash of the chunk.
- Calculate ASCON hash of the chunk.
- Save the calculated hashes along with metadata (e.g., file name, chunk index) into the SQL database.
- Index the SQL database for efficient retrieval.

5.2. Function Process_Directory(directory_path: str)

- List all PE files in the specified directory.
- For each PE file in the directory, call analyze_PE_file function.

5.3. Function Compare_with_Database(file_path: str)

- Read the PE file specified by file_path.
- Extract the .text section of the PE file.
- Divide the .text section into chunks of 10KB.

For each chunk:

- Calculate SHA-256 hash of the chunk.
- Calculate SHA-3 hash of the chunk.
- Calculate BLAKE2s hash of the chunk.
- Calculate Skein hash of the chunk.
- Calculate MD5 hash of the chunk.
- Calculate ASCON hash of the chunk.
- Query the SQL database for matching hashes.
- If a match is found, log or report the match along with relevant metadata.

5.4. Main Program

- Accept user input for either analyzing a single PE file or processing a directory.
- If analyzing a single PE file, call analyze_PE_file function.
- If processing a directory, call process_directory function.
- If comparing with the database, call compare_with_database function.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 August 2024 doi:10.20944/preprints202408.0263.v1

6. Results and Discussion

Our comprehensive analysis of over a thousand malware samples utilizing ASCON-based fuzzy hashing emphasizes its superiority in terms of speed and efficiency. While it is designed for speed and security, its performance was also relative to the capabilities of other algorithms, as seen in the table.

ASCON-Hash, within the range of 224 to 512 bits output size, typically demonstrated 4-6 cycles per byte. Its design for speed and security played out in the field, making it a formidable baseline standard for hashing speeds. It looked particularly efficient when contrasted with the other hashing algorithms.

SHA-256, a widely used and standardized algorithm, showed a 1.33x - 2x slower performance than ASCON. It underwent 8-12 typical cycles per byte with a singular output size of 256 bits. SHA-3, the NIST SHA-3 competition winner known for its good security, exhibited a markedly slower speed ranging from 2x - 2.67x slower than ASCON despite having the same output size as SHA-256.

Though BLAKE2s has been noted for being fast and secure, especially for its use in cryptocurrencies, it showed a 1x - 1.33x slower performance than ASCON. BLAKE2s carries out 6-8 typical cycles per byte and has an output size of 256 bits. Skein, attributed with a high degree of configurability and a focus on security, performed 1.67x - 2.33x slower than ASCON.

Interestingly, the oldest algorithm in the lot, MD5, demonstrated 0.67x - 1x faster speed than ASCON but at the glaring cost of security. This legacy algorithm with known vulnerabilities operates with beats of 4-6 cycles per byte and hands out a lower output size of 128 bits.

In conclusion, the results illustrate the balance ASCON offers between speed and security that many other hashing algorithms do not appear to match. Its performance serves as a baseline to validate the efficiency of other hashing functions, revolutionising malware analysis in terms of speed and security, hence marking it as a fitting choice for real-world applications.

Algorithm	Output Size (bits)	Typical Cycles per Byte	Relative Speed to ASCON
ASCON-Hash	224-512	4-6	1x
SHA-256	256	8-12	1.33x - 2x slower
SHA-3	256	10-13	2x - 2.67x slower
BLAKE2s	256	6-8	1x - 1.33x slower
Skein	256-512	10-15	1.67x - 2.33x slower
MD5 (legacy)	128	4-6	0.67x - 1x faster (insecure)

Figure 3. Hashing Results

7. Future Work

In future we can develop a GUI-based tool with linked database for long-term memory for historical tracking of malware versions as well inducing visual charts for better threat intelligence. In addition, we can integrate machine learning algorithms to classify different malware samples based on hashes of different sections of the sample.

 doi:10.20944/preprints202408.0263.v1

References

1. Li, Y.; Sundaramurthy, S.C.; Bardas, A.G.; Ou, X.; Caragea, D.; Hu, X.; Jang, J. Experimental study of fuzzy hashing in malware clustering analysis. 8th workshop on cyber security experimentation and test (cset 15), 2015.
2. Rodriguez-Bazan, H.; Sidorov, G.; Escamilla-Ambrosio, P.J. Android Ransomware Analysis Using Convolutional Neural Network and Fuzzy Hashing Features. *IEEE Access* **2023**.
3. Shiel, I.; O'Shaughnessy, S. Improving file-level fuzzy hashes for malware variant classification. *Digital Investigation* **2019**, *28*, S88–S94.
4. Tariq, U.; Ahmed, I.; Khan, M.A.; Bashir, A.K. Fortifying IoT against crimpling cyber-attacks: a systematic review. *Karbala International Journal of Modern Science* **2023**, *9*, 9.
5. Aghili, S.F.; Sedaghat, M.; Singelée, D.; Gupta, M. MLS-ABAC: Efficient multi-level security attribute-based access control scheme. *Future Generation Computer Systems* **2022**, *131*, 75–90.
6. Naik, N.; Jenkins, P.; Savage, N. A ransomware detection method using fuzzy hashing for mitigating the risk of occlusion of information systems. 2019 international symposium on systems engineering (ISSE). IEEE, 2019, pp. 1–6.
7. Naik, N.; Jenkins, P.; Savage, N.; Yang, L.; Boongoen, T.; Iam-On, N. Fuzzy-Import Hashing: A malware analysis approach. 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). IEEE, 2020, pp. 1–8.
8. Naik, N.; Jenkins, P.; Savage, N.; Yang, L.; Boongoen, T.; Iam-On, N.; Naik, K.; Song, J. Embedded YARA rules: strengthening YARA rules utilising fuzzy hashing and fuzzy rules for malware analysis. *Complex & Intelligent Systems* **2021**, *7*, 687–702.
9. Naik, N.; Jenkins, P.; Savage, N.; Yang, L.; Naik, K.; Song, J.; Boongoen, T.; Iam-On, N. Fuzzy hashing aided enhanced YARA rules for malware triaging. 2020 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2020, pp. 1138–1145.
10. Chang, D.; Hong, D.; Kang, J.; Turan, M.S. Resistance of Ascon Family against Conditional Cube Attacks in Nonce-Misuse Setting. *IEEE Access* **2022**, *11*, 4501–4516.
11. Mostafa, A.M.; Ezz, M.; Elbashir, M.K.; Alruily, M.; Hamouda, E.; Alsarhani, M.; Said, W. Strengthening Cloud Security: An Innovative Multi-Factor Multi-Layer Authentication Framework for Cloud User Authentication. *Applied Sciences* **2023**, *13*, 10871.
12. Pagani, F.; Dell'Amico, M.; Balzarotti, D. Beyond precision and recall: understanding uses (and misuses) of similarity hashes in binary analysis. Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy, 2018, pp. 354–365.
13. Namanya, A.P.; Awan, I.U.; Disso, J.P.; Younas, M. Similarity hash based scoring of portable executable files for efficient malware detection in IoT. *Future Generation Computer Systems* **2020**, *110*, 824–832.
14. Ali, H.; Batool, K.; Yousaf, M.; Islam Satti, M.; Naseer, S.; Zahid, S.; Gardezi, A.A.; Shafiq, M.; Choi, J.G. Security Hardened and Privacy Preserved Android Malware Detection Using Fuzzy Hash of Reverse Engineered Source Code. *Security & Communication Networks* **2022**.
15. Samra, A.A.A.; Qunoo, H.N.; Al-Rubaie, F.; El-Talli, H. A survey of static android malware detection techniques. 2019 IEEE 7Th palestinian international conference on electrical and computer engineering (PICECE). IEEE, 2019, pp. 1–6.
16. Aslan, Ö.A.; Samet, R. A comprehensive review on malware detection approaches. *IEEE access* **2020**, *8*, 6249–6271.
17. Amira, A.; Derhab, A.; Karbab, E.B.; Nouali, O. A survey of malware analysis using community detection algorithms. *ACM Computing Surveys* **2023**, *56*, 1–29.