Distinguishing Tor From Other Encrypted Network Traffic Through Character Analysis

Pitpimon Choorod[1], Tobias J. Bauer[2], and Andreas Aßmuth[3]
[1]King Mongkut's University of Technology North Bangkok, Prachinburi, Thailand
Email: pitpimon.c@itm.kmutnb.ac.th
[2]Fraunhofer Institute for Applied and Integrated Security, Weiden, Germany
Email: tobias.bauer@aisec.fraunhofer.de
[3]Ostbayerische Technische Hochschule Amberg-Weiden, Amberg, Germany
Email: a.assmuth@oth-aw.de

***Abstract*—For journalists reporting from a totalitarian regime, whistleblowers and resistance fighters, the anonymous use of cloud services on the Internet can be vital for survival. The Tor network provides a free and widely used anonymization service for everyone. However, there are different approaches to distinguishing Tor from non-Tor encrypted network traffic, most recently only due to the (relative) frequencies of hex digits in a single encrypted payload packet. While conventional data traffic is usually encrypted once, but at least three times in the case of Tor due to the structure and principle of the Tor network, we have examined to what extent the number of encryptions contributes to being able to distinguish Tor from non-Tor encrypted data traffic.**

***Keywords*—Anonymization; Tor; encryption.**

I. Introduction

When it comes to security for cloud services, most people first think of ensuring the security goals of confidentiality, integrity, availability, and authenticity. However, anonymization services also play an important role, as they are used, for example, by journalists or opponents of regimes in authoritarian countries to access cloud services and to provide information about grievances in the country in question. In general, anonymization services can increase the privacy of users of cloud services, as anonymization prevents unauthorised third parties from tracking or profiling users based on their cloud-related activities. To be fair, it must also be noted at this point that criminals also have an interest in anonymization services, whether to conceal their criminal activities or to set up and operate largely anonymous trading platforms on the Darknet, e.g., Silkroad [1].

The Tor project (Tor, short for The Onion Router) has been a very popular and free anonymization service on the Internet for years. The basic idea of anonymization can be described as follows: a number of n nodes of the Tor network are identified via which communication is to take place, for example accessing a website via http. Depending on the number n (the default is $n = 3$), the actual request is encrypted n times in succession, creating n (encryption) layers – like an onion. Each node of the identified path through the Tor network now removes one of these layers by decrypting it before forwarding the data to the next Tor node. The last layer is finally removed by the exit node, whose IP address is then visible when accessing the actual website, but not the IP address of the actual user's computer. Each node in the Tor network only knows its predecessor and its successor for the respective path. For a detailed description of how Tor works, please refer to [2] and, of course, to the documentation of the Tor project [3].

Against the background described above, the question can now be asked whether and how it is possible to distinguish Tor traffic from otherwise encrypted traffic when monitoring network traffic. This question has a fundamental core aspect for cryptography. The definition of perfect secrecy goes back to Shannon [4]. The necessary and sufficient condition for perfect secrecy is $\Pr(C = c \,|\, M = m) = \Pr(C = c)$, where $\Pr(C = c)$ is the (a priori) probability of obtaining the ciphertext c, and $\Pr(C = c \,|\, M = m)$ is the conditional probability of ciphertext c if message m was chosen for encryption. Building on this theorem, modern textbooks on cryptography describe experiments that are the basis for security definitions for encryption schemes as we use them today. As an example of such an experiment, the so-called adversarial indistinguishability experiment for probabilistic symmetric-key encryption schemes is presented here (according to [5]):

1) An attacker $\mathcal{A}$ chooses two messages m_0 and m_1 of the same length for a given encryption scheme with security parameter N. The security parameter may be viewed as corresponding to the length of the key.
2) A random key k is generated (depending on N) and a bit $b \in \{0, 1\}$ is chosen at random. $\mathcal{A}$ receives the so-called challenge ciphertext $c \leftarrow \mathrm{Enc}_k(m_b)$.
3) $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$.
4) The result of the experiment is 1 if $b = b'$, otherwise 0.

In the case of perfect secrecy according to Shannon's theorem, the result of the experiment corresponds to the guess probability of $50\,\%$. For security definitions for modern encryption schemes, the probability for result 1 must be increased slightly, whereby this increase is set via the security parameter N and its actual value is a negligible function in N for all realistic adversaries. Now imagine running this experiment in parallel for two different encryption schemes, where in 1) the length of all messages is the same, and the result of both experiments is only slightly more than $50\,\%$ for all realistic adversaries – if $\mathcal{A}$ cannot practically decide which of the messages led to c

 ISBN: 978-1-68558-156-5

by encryption, can $\mathcal{A}$ distinguish which challenge ciphertext was generated by which encryption scheme? According to Rogaway, several modes of operations for modern blockciphers achieve computational indistinguishability from random bits [6]. So, if $\mathcal{A}$ cannot distinguish any ciphertext from a random bitstring of the same length, it should not be feasible to distinguish Tor-encrypted network traffic from otherwise (non-Tor) encrypted network traffic.

The paper is structured in the following manner: Section II provides an overview of published work that deals with the distinction between Tor and non-Tor encrypted traffic. Section III summarises the most important results of a novel approach for classifying Tor and non-Tor traffic presented by Pitpimon Choorod in her PhD thesis [7]. Based on these results, new experiments have been carried out which are presented in Section IV. Finally, Section V ends with a conclusion and an outlook on future work.

II. Related Work

The robustness of encryption schemes has led researchers to study the Tor traffic classification domain using flow-based or packet-based features. Lashkari et al. [8], the creators of the University of New Brunswick, Canadian Institute for Cybersecurity (UNB-CIC) dataset, achieved high performance in detecting Tor traffic using time-based features and attained precision and recall rates above 0.9 with the C4.5 algorithm. Using the same dataset, Kim et al. [9] instead focused on payload-based features with the first 54 bytes of TCP packet headers as input. The results indicated that the one-dimensional convolutional neural network model outperformed the C4.5 algorithm of [8], achieving precision and recall rates of 1.0 in classifying both Tor and non-Tor traffic. Hu et al. [10] expanded the scope of Darknet traffic analysis. They distinguished four Darknet traffic types including Tor, I2P, ZeroNet, and Freenet using 26 time-based flow features, achieving an accuracy of $96.9\,\%$. However, while flow features are effective in classifying Tor traffic, factors such as network sensitivities, including asymmetric routing, can undermine the reliability of time-based features. The approach chosen in [7] addresses this limitation by enhancing reliability under diverse network conditions. We focus on extracting non-timing related features from the encrypted data within packet payloads, thereby presenting a challenge to the conventional assumptions of Shannon's theorem.

III. Preliminary Work

This section describes the preliminary work that Pitpimon Choorod carried out as part of her PhD thesis [7]. A publication summarising the key aspects of the PhD thesis is also available [11].

In computer networks, data payloads are commonly represented as hexadecimal characters, using a base-16 numbering system that ranges from *0* to *9* and *a* to *f*. The study focused on analysing these hexadecimal characters in their single-digit (1-hex) form extracted from encrypted data. To facilitate the analysis, two key statistical features were used: 1) a frequency set feature, which consists of 16 individual features for quantifying the occurrence of each hexadecimal character within data payloads, and 2) a frequency ratio set, also including 16 features, for calculating the proportion of each character's frequency relative to the total character count within a payload. The normalisation of these frequencies was crucial to ensure length normalisation, thereby minimising potential biases in analysing encrypted payloads that could arise from relying solely on their absolute packet lengths.

The analysis utilised two data sources to validate the robustness and reliability of the results. The first was a public Tor dataset from the UNB-CIC, where network traffic was categorised into eight application types (audio, browsing, chat, email, FTP, P2P, video, and VoIP). In addition to the public dataset, a private dataset was created by capturing Tor-encrypted traffic data packets using Wireshark. The corresponding data consists of browsing applications. Table I presents the number of instances for the eight application types in the public dataset and one application type in the private dataset. It should be noted that the instances for both Tor and non-Tor are balanced.

TABLE I
Number of balanced Tor and non-Tor instances for nine applications

Audio	26,082	Email	12,300	Video	32,154
Browsing	71,950	FTP	514,952	VoIP	737,382
Chat	6,504	P2P	433,770	Private	29,600

According to Section I, the investigation commenced with the assumption that there is no difference between Tor and non-Tor traffic in terms of encrypted payloads. Initially, descriptive statistics were utilised to describe and summarise the characteristics of the sample data. In this study, statistical measurements including character distribution were employed, which helps reveal patterns in how individual features are spread across the range of values in both Tor and non-Tor encrypted payloads. The mean measurement indicates the central tendency of each feature, aiding in identifying the average value of the individual features. Standard deviation measures the variability or dispersion of each feature, highlighting trends or patterns. Minimum and maximum values provide insights into the range of features within the Tor and non-Tor datasets, with a wider range potentially indicating greater variability in traffic characteristics. The results clearly showed that all measurements of Tor and non-Tor were significantly different, except for the ratio features. This exception can be attributed to the effect of normalisation, which tends to minimise discrepancies in data scale and distribution, thereby making the ratio features appear more similar across both datasets. Additionally, these findings were generalised using the Mann-Whitney test, which revealed a significant differentiation rate between Tor and non-Tor traffic of $95.42\,\%$ for the public dataset and $100\,\%$ for the private dataset.

The second phase of the study, run in Weka [12], focused on classifying Tor and non-Tor traffic using machine

 ISBN: 978-1-68558-156-5

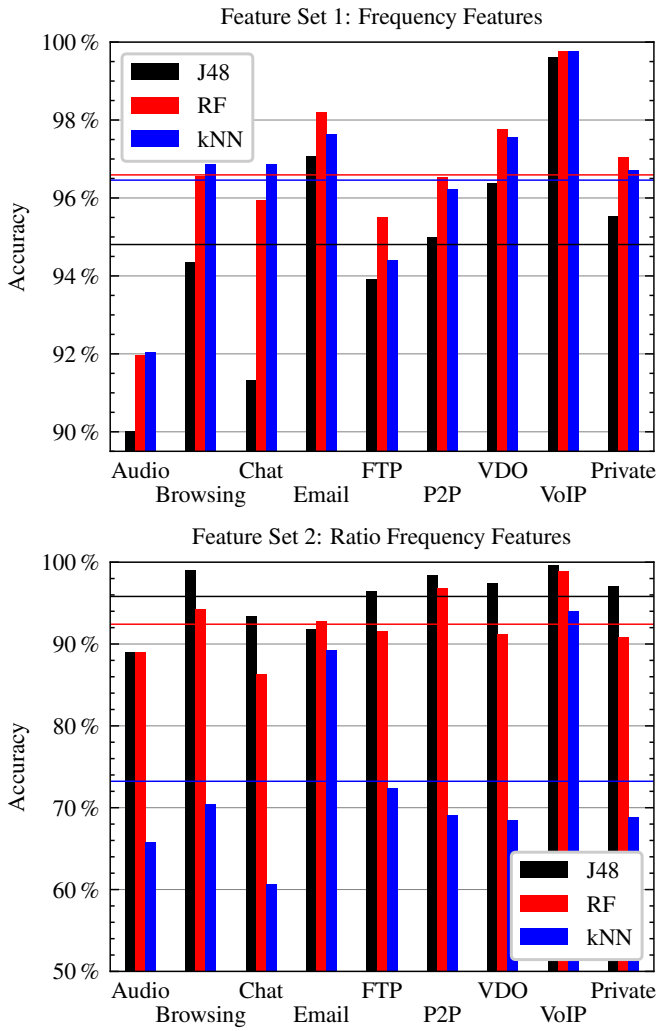

Figure 1. Results of the approach proposed in [7].

learning. Three supervised learning algorithms were used: J48 [13], Random Forest (RF) [14], and k-Nearest Neighbors (kNN) [15], with a specific focus on encrypted payload features. The J48 algorithm is identical to the aforementioned C4.5 algorithm. As depicted in Figure 1, for the frequency feature set on the public dataset, it can be noted that classification accuracy exceeded 90 % for all models across all applications. Notably, both RF and kNN achieved the highest score of 99.77 % for VoIP. For the private dataset, RF demonstrated superior performance with a score of 97.06 %.

Regarding the ratio frequency feature set on public datasets, all models surpassed a classification accuracy of 60.62 % across all applications, with J48 achieving the best score of 99.63 % for VoIP. For the private dataset, J48 reached an impressive accuracy of 97.12 %. These results show a similar trend in both public and private datasets, ensuring the consistency of these findings.

These results conclusively demonstrate that Tor and non-Tor traffic are statistically distinct, enabling efficient classification of both types of traffic using features derived exclusively from a single encrypted payload packet.

IV. New Experiments

One might be tempted to explain these results by the fact that encryption in Tor and non-Tor encrypted traffic in practice does not show the desired property presented in Section I. If we compare Tor to non-Tor encrypted traffic, the main difference is that while, e.g., TLS traffic is encrypted only once Tor traffic is encrypted multiple times because of the onion-like layer model. Therefore, in this paper we focus solely on distinguishing between single-encrypted data and triple-encrypted data. We demonstrate that data that was encrypted one time has the same statistical properties as data that was encrypted three times. This results in machine learning algorithms being unable to distinguish between them. We have tested the Advanced Encryption Standard (AES) algorithm in various modes of operation.

As a first step, sample data needed to be generated. In order to study the effect of the underlying data, our generation method outputs two sets of data samples. Each set contains $\# = 10^6$ strings of $l = 512$ bytes. The length l was chosen to be a multiple of the AES block size of 128 bits and following the message length in the Tor specification [16]. The first set is generated using the cryptographically secure pseudorandom number generator */dev/urandom* of Linux, whereas the second set contains the same amount of samples, but each sample is a string of null-bytes, representing data with zero randomness to it. We denote a sample of the first set as r_i^0 (random data) and a sample of the second set as z_i^0 (zeros) with $0 \leq i < \#$.

Next, we generate $\#$ initialization vectors (IVs) $\mathrm{iv}_i^1, \mathrm{iv}_i^2, \mathrm{iv}_i^3$ and encryption keys $\mathrm{k}_i^1, \mathrm{k}_i^2, \mathrm{k}_i^3 \quad (0 \leq i < \#)$ at random. The encryption algorithm takes any data d, an IV iv, and an encryption key k and outputs a ciphertext $c = \mathrm{Enc}\,(d, \mathrm{iv}, \mathrm{k})$. We then perform a single encryption of samples r_i^0 and z_i^0 to obtain r_i^1 and z_i^1. Next, two more rounds of encryption are performed to obtain r_i^3 and z_i^3, respectively. The following equations illustrate the process:

$$r_i^1 = \mathrm{Enc}\left(r_i^0, \mathrm{iv}_i^1, \mathrm{k}_i^1\right) \tag{1}$$

$$z_i^1 = \mathrm{Enc}\left(z_i^0, \mathrm{iv}_i^1, \mathrm{k}_i^1\right) \tag{2}$$

$$r_i^3 = \mathrm{Enc}\left(\mathrm{Enc}\left(r_i^1, \mathrm{iv}_i^2, \mathrm{k}_i^2\right), \mathrm{iv}_i^3, \mathrm{k}_i^3\right) \tag{3}$$

$$z_i^3 = \mathrm{Enc}\left(\mathrm{Enc}\left(z_i^1, \mathrm{iv}_i^2, \mathrm{k}_i^2\right), \mathrm{iv}_i^3, \mathrm{k}_i^3\right) \tag{4}$$

We denote the sets of samples by their capital letter, i.e.,

$$R^1 = \left\{r_1^1, r_2^1, \ldots, r_\#^1\right\}, \qquad R^3 = \left\{r_1^3, r_2^3, \ldots, r_\#^3\right\},$$
$$Z^1 = \left\{z_1^1, z_2^1, \ldots, z_\#^1\right\}, \qquad Z^3 = \left\{z_1^3, z_2^3, \ldots, z_\#^3\right\}.$$

In order to study the effect of different AES modes of operation, we perform these preparatory steps for each mode. In total, we opted to study these three modes: Cipher Block Chaining (CBC), Counter (CTR), and Electronic Codebook (ECB). The CBC mode is widely used within the TLS 1.2 specification [17] and the CTR mode forms the basis for the Galois/Counter Mode (GCM) [18], which is used extensively throughout the Internet [19]. Furthermore, GCM is used in two out of five specified cipher suites of TLS 1.3 [20] and is preferred by a majority of web servers [19]. In addition to that, we include a third mode of operation, ECB, in our experiments. This mode of operation is highly insecure as

 ISBN: 978-1-68558-156-5

it leaks the equality of blocks [6], does not provide the required randomization of the ciphertext, and should therefore not be used within any cryptographic protocols. However, we can show that even this mode of operation achieves the indistinguishability between single-encrypted data and triple-encrypted data provided that a random key is used.

Our experiments follow a simple four-step procedure:

1) Generate a dataset with features $\mathcal{X}$ and labels $\mathcal{Y}$:
$\mathcal{D} = \{d_1, d_2, \ldots, d_{2\cdot\#}\} \overset{\text{e.g.}}{=} R^1 \cup R^3$
$\mathcal{X} = \{X_1, X_2, \ldots, X_{2\cdot\#}\}, \quad X_i = F(d_i)$
$\mathcal{Y} = \{y_1, \ldots, y_{2\cdot\#}\} = \begin{cases} 0 & \text{if } d_i \text{ single-encrypted} \\ 1 & \text{if } d_i \text{ triple-encrypted} \end{cases}$.
2) Split $(\mathcal{X}, \mathcal{Y})$ into a training set $(\mathcal{X}_{\text{tr}}, \mathcal{Y}_{\text{tr}})$ and a test set $(\mathcal{X}_{\text{te}}, \mathcal{Y}_{\text{te}})$ using a 75:25-split.
3) Fit a machine learning model to the training set.
4) Evaluate the trained machine learning model on the test set and compute a confusion matrix.

The function F denotes the feature engineering, which is similar to the previous work [7]. $F(d_i)$ simply counts the hexadecimal digits *0* to *f* and returns the relative frequencies for each of the 16 digits, i.e., a vector $X_i \in \mathbb{R}^{16}$. Since the original data strings are fixed-length strings of random data $\left(r_i^0\right)$ or zeros $\left(z_i^0\right)$, relative and absolute frequencies behave identically, which is why we used only the relative frequencies of the hexadecimal digits.

The results are depicted in Figures 2, 3, and 4. Each figure shows the results for one mode of operation using three machine learning algorithms – Random Forest (RF), Decision Tree (DT), and k-Nearest Neighbors (kNN) – on two datasets $\mathcal{D}_Z = Z^1 \cup Z^3$ (upper row) and $\mathcal{D}_R = R^1 \cup R^3$ (lower row). The accuracy value of each experiment is displayed in the sub-plot title. In order to ensure comparability with the preliminary work described in Section III, we opted to employ the same three machine learning algorithms. However, in this paper we use the *scikit-learn* [21] machine learning framework. This framework uses for Decision Tree construction the CART algorithm that is similar to C4.5 and J48, respectively [22].

Our results show clearly that none of these machine learning models is able to distinguish between single-encrypted and triple-encrypted payload using the relative frequencies of the 16 hexadecimal digits as feature vectors. The accuracy is always about $50\,\% \pm 0.17\,\%$, which is due to run-to-run variance and does not indicate any ability to distinguish these two categories. We would like to remind the reader that $50\,\%$ is exactly the guess probability. Even the insecure ECB mode of operation achieves the indistinguishability property described in Section I and the machine learning models are therefore unable to predict the class correctly in significantly more than $50\,\%$ of all cases (cf. Figure 2).

These new experiments clearly indicate that the distinction between one-time and three-time encryption cannot be the decisive criterion in the generation of ciphertexts. Therefore, the reason to why the method described in [7] and abridged in Section III is able to distinguish Tor from non-Tor encrypted data traffic with such high rates must not be related to the number of encryption passes.

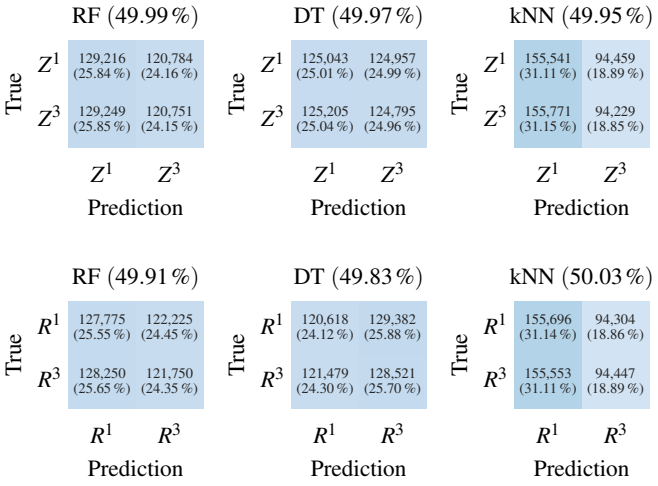

Figure 2. Results with the ECB mode of operation.

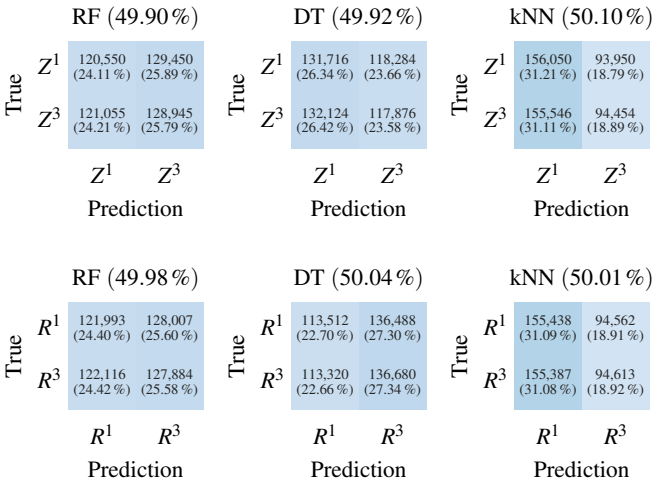

Figure 3. Results with the CBC mode of operation.

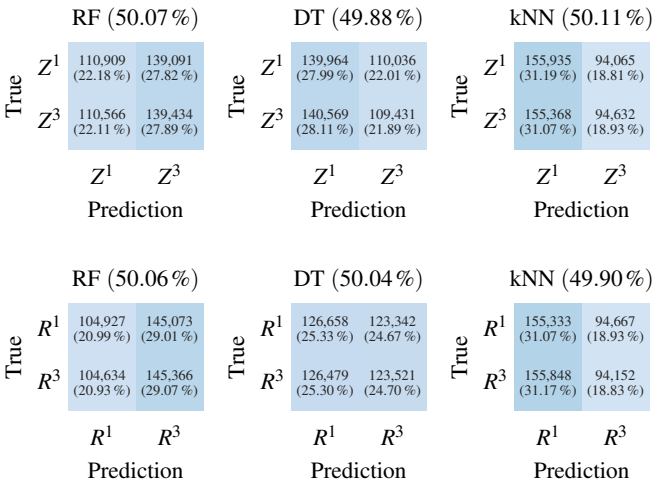

Figure 4. Results with the CTR mode of operation.

V. Conclusion and Future Work

In her doctoral thesis, Pitpimon Choorod presented a method which allows to distinguish Tor and non-Tor encrypted data traffic at high rates only on the basis of the analysis of hex digits occurring in a single encrypted data packet or their relative frequency. However, it is still not fully understood, why this is possible. One might think that this distinction is made possible by the fact that Tor traffic, unlike other encrypted traffic, is encrypted multiple times, but this would be in clear contradiction to the cryptographic theory of secure encryption. In this paper, we have deliberately omitted the technical network superstructure and concentrated solely on the distinction between single- and triple-encrypted data traffic, whereby we have also examined different operating modes for the AES block cipher. The results are absolutely clear: with the proposed method none of the the three machine learning algorithms, Random Forest, Decision Tree, or k-Nearest Neighbor, is capable of distinguishing between single- and triple-encrypted data. These results are in accordance with crypto theory and illustrate that encryption is not the reason why a distinction can be made between Tor and non-Tor encrypted traffic.

In order to better understand why this distinction is nevertheless possible, we will conduct further experiments in the future to gradually rule out possible explanations and identify the actual cause.

References

[1] R. Liggett, J. R. Lee, A. L. Roddy, and M. A. Wallin, "The dark web as a platform for crime: An exploration of illicit drug, firearm, csam, and cybercrime markets," in *The Palgrave Handbook of International Cybercrime and Cyberdeviance*, T. J. Holt and A. M. Bossler, Eds. Cham: Springer International Publishing, 2020, pp. 91–116, ISBN: 978-3-319-78440-3. DOI: 10.1007/978-3-319-78440-3_17.

[2] R. Dingledine, N. Mathewson, and P. F. Syverson, "Tor: The second-generation onion router," in *USENIX Security Symposium*, 2004. [Online]. Available: https://api.semanticscholar.org/CorpusID:8274154 (visited on 2024-03-26).

[3] The Tor Project, Inc. "Tor Project Website." (), [Online]. Available: https://www.torproject.org/ (visited on 2024-03-26).

[4] C. E. Shannon, "Communication theory of secrecy systems," *The Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949. DOI: 10.1002/j.1538-7305.1949.tb00928.x.

[5] J. Katz and Y. Lindell, *Introduction to Modern Cryptography, 3rd Edition*. New York: Chapman and Hall/CRC, 2020, ISBN: 9781351133036. DOI: 10.1201/9781351133036.

[6] P. Rogaway, "Evaluation of some blockcipher modes of operation," University of California, Davis, Dept. of Computer Science, Technical Report, Feb. 10, 2011. [Online]. Available: https://web.cs.ucdavis.edu/~rogaway/papers/modes.pdf (visited on 2024-03-26).

[7] P. Choorod, "Classifying tor traffic using character analysis," Ph.D. dissertation, University of Strathclyde, Department of Computer and Information Sciences, 2023.

[8] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," in *International Conference on Information Systems Security and Privacy*, SciTePress, vol. 2, 2017, pp. 253–262.

[9] M. Kim and A. Anpalagan, "Tor traffic classification from raw packet header using convolutional neural network," in *2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII)*, IEEE, 2018, pp. 187–190.

[10] Y. Hu, F. Zou, L. Li, and P. Yi, "Traffic classification of user behaviors in tor, i2p, zeronet, freenet," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2020, pp. 418–424.

[11] P. Choorod, G. Weir, and A. Fernando, "Classifying tor traffic encrypted payload using machine learning," *IEEE Access*, vol. 12, pp. 19 418–19 431, 2024. DOI: 10.1109/ACCESS.2024.3356073.

[12] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, *The WEKA data mining software: an update*. ACM New York, NY, USA, 2009, vol. 11, pp. 10–18.

[13] N. Bhargava, G. Sharma, R. Bhargava, and M. Mathuria, "Decision tree analysis on j48 algorithm for data mining," *Proceedings of international journal of advanced research in computer science and software engineering*, vol. 3, no. 6, 2013.

[14] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[15] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN model-based approach in classification," in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*, Springer, 2003, pp. 986–996.

[16] The Tor Project, Inc. "Tor Specifications – Preliminaries." (2024), [Online]. Available: https://spec.torproject.org/tor-spec/preliminaries.html (visited on 2024-03-26).

[17] E. Rescorla and T. Dierks, *The Transport Layer Security (TLS) Protocol Version 1.2*, RFC 5246, Aug. 2008. DOI: 10.17487/RFC5246. [Online]. Available: https://www.rfc-editor.org/info/rfc5246.

[18] M. J. Dworkin, "Recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac," Tech. Rep., Nov. 2007. DOI: 10.6028/nist.sp.800-38d.

[19] D. Warburton and S. Vinberg. "The 2021 TLS Telemetry Report." (Oct. 20, 2021), [Online]. Available: https://www.f5.com/labs/articles/threat-intelligence/the-2021-tls-telemetry-report (visited on 2024-03-26).

[20] E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.3*, RFC 8446, Aug. 2018. DOI: 10.17487/RFC8446. [Online]. Available: https://www.rfc-editor.org/info/rfc8446.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[22] scikit-learn developers. "1.10. Decision Trees – scikit-learn." (2024), [Online]. Available: https://scikit-learn.org/stable/modules/tree.html (visited on 2024-03-26).

 ISBN: 978-1-68558-156-5