Using Machine Learning to Detect the File Compression or Encryption

Nica Ameeno[1], Kalib Sherry[2], Kanwalinderjit Gagneja[3]
[1,2,3]Florida Polytechnic University, Lakeland, FL, USA
[1]naimino0723@floridapoly.edu, [2]cshirley@floridapoly.edu, [3]kgagnej@floridapoly.edu

Abstract:
The detection of file type and their features can be difficult and will have a continuously increasing urgency in the computer science and cyber security field. File type and feature detection includes but not limited to file extensions, compression, encryption, and the presence/absence of bytes. There are also ways to monitor the contents of the file and is becoming more popular. In this paper, a "proof of concept" is demonstrated with experiments in which file type and features can be detected and clustered for comparison. Number of ways are recommended to tell the difference between encryption and compression of files.

Keywords: Cryptography, Compression, Encryption, RC4, Machine Learning, ZIP, Supervised.

1. Introduction

There is an increasing importance regarding data security and privacy. Technology now a days needs file type detection. Identifying a file's file type is imperative when dealing with suspicious or unfamiliar data. Companies that use computers to transmit large number of files across networks need file detection or monitoring techniques to prevent compromising their network and data integrity [23], [24].

2. Cryptography

Cryptography is the study of techniques for ensuring the secrecy and/or authenticity of information [1]. Cryptography is defined as the science of protecting data [26]. This scienceofferstechniques to convert data into unreadable form, so that lawfuluser be able to access information at the other end [3]. There are a couple of generic types of encryption:

(a) *Symmetric Encryption:* This encryption algorithms use the same key for both encryption and decryption. It is also considered symmetric if the decryption key can be derived from the encryption key [27]. Figure 1 shows a simple example of symmetric encryption.

(b) *Asymmetric encryption:* This encryption is present when the algorithm uses different keys for encrypting and decrypting data (as shown in Figure 2) [25], [5]. In this paper, symmetric encryption methods are used to encrypt the sample data files.

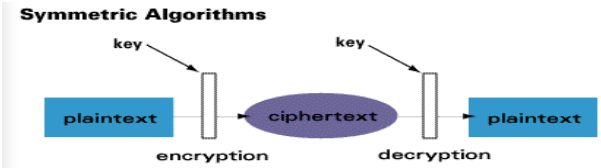

Fig. 1. Symmetric Encryption

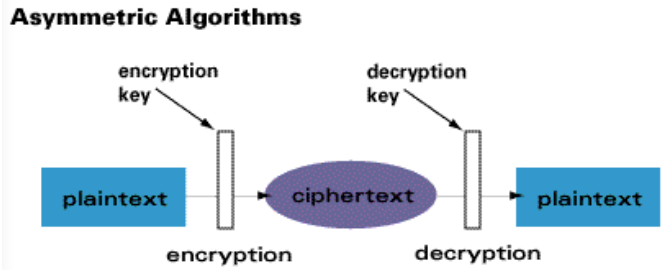

Fig. 2. Asymmetric Encryption

2.1 Cryptographic Methodologies

2.1.1 RC4

The RC4 cipher was designed by Ron Rivest in 1987. It was originally used for securing Web traffic and wireless traffic [4]. Currently, it is used in WEP and WPA protocols that are most commonly seen on wireless routers. The algorithm is based on random permutation. It is an extremely simple and quick stream cipher to implement [12]. To generate a key stream, all 256 possible bytes must first be permuted. The typical key length is somewhere between 40 and 256 bits [13]. It is no longer considered a secure way to encrypt data; however, this method of encryption is used due to its simplicity.

2.1.2 RSA

RSA is another method of encryption that is used to securely transmit messages over the internet [16], [20], [21]. This encryption method is considered to bemaking it difficult to factorize large integers [9]. The public key is combination of numbers. One number is the multiplication of two large prime numbers. The private key is resultof the same two prime numbers [6]. The strength of the encryption is dependent on the size of the key [8]. That means as the length of the key size increases, then the level of security increases. Figure 3 shows the basics of how RSA encryption works across a network.

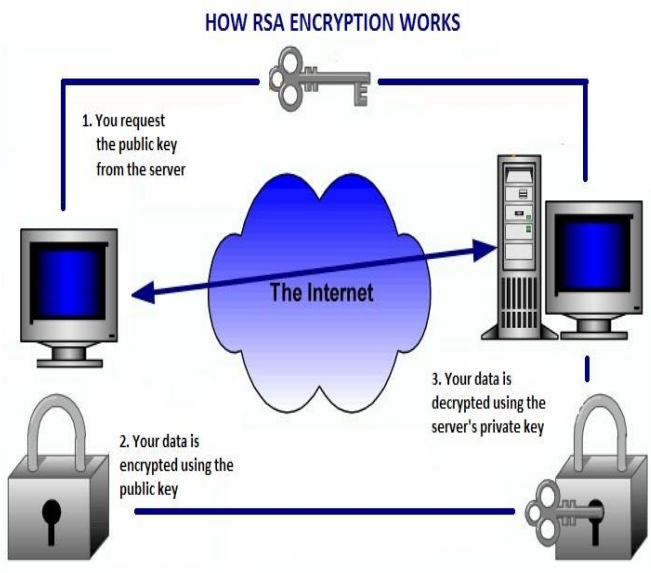

Fig. 3. RSA Encryption

2.1.3 DES

The Data Encryption Standard (DES) is a symmetric-key block cipher that uses 56-bit key [10]. Because it is a block cipher, it is able to take plaintext broken into fixed length blocks which are encrypted simultaneously. It processes the message blocks through 16 rounds in order to encrypt them.

3. AES

AES stands for Advanced Encryption Standard and is symmetric encryption algorithm. This algorithm is 6 times faster than triple DES. AES is designed to overcome lack of computing power. It is also more secure than DES using a potential key size of 128/192/256-bits. Unlike DES, the number of rounds needed to encrypt is dependent on the length of the key [11]. AES requires at least 10 iterations for a key of length 128 bit, 12 iterations for a key of length of 192-bit, and 14 iterations for a key of length of 256-bit.

4. Compression

File compression removes redundant data from a file(s) by replacing it with smaller variables. One example of this is replacing words in a given text file with short identifiers. A text file with the words "Digital Forensics Digital" may be replaced by the identifiers D2F1. This takes up less space than the words "Digital Forensics Digital". This method is called lossless and fully restored the file's original state [19]. There is another method called lossy which can produce much smaller compressed file sizes, but there is greater quality loss upon decompressing the file. This can also work with binary files by replacing repeated binary patterns allowing a significantly reduced file size. The main goal for compressing files is to free up space and allow easier file transport.

4.1 Compression File Types

4.1.1 ZIP

ZIP files are standard compressed files designed for cross-platform data exchange. It supports multiple different compression algorithms. ZIP files are identified by metadata consisting of defined record type containing the storage information necessary for maintaining the files placed into it [7].

4.1.2 RAR

RAR files are archives that contains compressed files using a RAR compression algorithm. It tends to compress data with a higher compression ratio than normal ZIP compression algorithms [28]. It can also create multi-volume archives split across several compressed files.

5. Machine Learning

In 1959 Arthur Samuel in his own words defined machine learning. According to him it is the study that gives computers the ability to learn shot of being unambiguously programmed. He defined the following two types of machine learning:

5.1 Supervised Machine Learning

Supervised machine learning is when the machine is fed data and is told what the data represents. In supervised machine learning you are essentially giving the input and the desired output. Let's say you have photos of fruit - apples, grapes, bananas, and cherries. You tell the machine that if it is big and red, it is an apple. If it is small and red, it is a cherry. If it is big and green, it is a banana. If it is small and green, it is a grape. When you give the machine a new photo of a fruit that is small and red, it will categorize it as a cherry. A couple of supervised learning algorithms include decision trees and Naive Bayesian.

5.1.1 Decision Trees

Decision trees are a method of classification based on the input and output of the data given to the machine. In decision analysis, it can be used to explain and represent the decisions and conclusions made by the machine. Figure 4 shows an accurate representation of a decision tree based on data given.

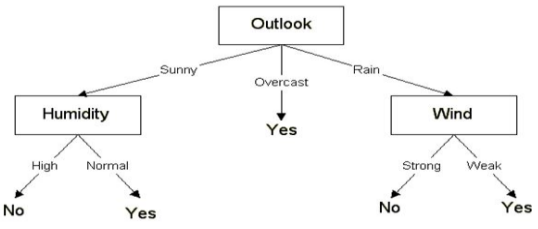

Fig. 4. Decision tree based on weather

5.1.2 Naive Bayesian

Naive Bayesian classifier calculates the probability for each factor given in the presented data. From there it reaches a conclusion that contains the highest probability of that result occurring. It is similar to how a decision tree function. The biggest difference is that is considers all the possibilities with the probabilities and reaches a conclusion whereas a decision tree only follows one path. Figure 5 below best shows an example of how Naive Bayesian works.

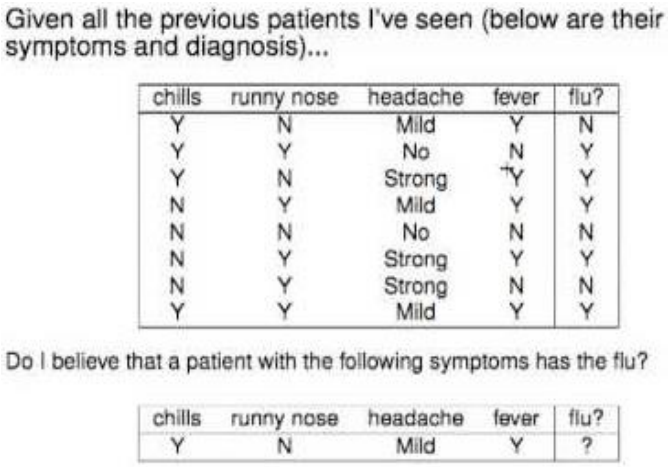

Fig. 5. Naive Bayesian classifier

5.2 Unsupervised Machine Learning

Unsupervised machine learning focuses more on clustering based on physical similarities of the given input. In supervised machine learning the machine is directed what is being inputted and what should be outputted. In unsupervised learning, the machine takes in the input and categorizes similar looking data. Using the fruit example again; let's say that you have the same fruit as before, except this time, the machine knows nothing about what the fruits are. It can only see what is given to it but doesnot know any information. The machine takes in the apple and cherry fruit and categorizes them as "Red" and categorizes the banana and grapes as "Green". The machine doesnot know they are apples, cherries, bananas, and grapes. It can only categorize them based on similarities present in the data. One type of unsupervised learning is K-means clustering.

5.2.1 K-means Clustering

K-means is a type of clustering used in unsupervised machine learning where the data given is unlabeled. The purpose of this algorithm is to find the groups in the data. Once the data has been graphed, a centroid(s), or central point(s), is selected and the data cluster is updated such that the closest data plots are focused around the nearest centroid. The process repeats until the data remains unchanged.

6. Implementation

In order to train machine learning algorithms to distinguish between compressed and encrypted data, a large amount of the work had to go into data collection and formatting. All the data used in this research was created using multiple programs and scripts on open-source books written in .txt files pulled from the Project Gutenberg website. And then formatted into one dataset using bash scripts and python.

6.1 File Compression

This step in the data manipulation process was given on Project Gutenberg. So one a script was written to download the first 105 files in the directory the initial data collection and file compression was completed.

6.2 File Encryption

Other script was written for file encryption on the files extracted from the zip files [14]. All files used the same password key "books" in the encryption in an attempt to make the encrypted data more uniform and easier to identify with our smaller dataset [15], [18], [22].

6.3 Data Collection

Once the compressed and encrypted files had all been collected, the binaries of those files were collected using hexdump. The line numbers and headers from hexdump's output where formatted out in order to make sure the algorithm couldnot remove them in training [19].

After all of the compressed and encrypted files binaries had been collected, they were formatted into one large and uniform .arff file. This was done by our python code that found the largest file, which therefore had the most features, and then appended zeros to the lines of every file until they all had the same uniform length. This meant that the dataset was 190 data points with over 1 million features each, creating 190 million unique features in the dataset. Then the arff header was generated and appended to the top of the file declaring the name and type of each feature. Then this was all written into an arff file to be later be transferred toweka.

6.4 Machine Learning Training and Testing

Once the dataset was created the file was ready to load into weka for training and testing, the only problem being that weka required a heap size of over 8GB in order to hold the data from the file.

Once the .arff file was loaded into weka all but the first 40,000 features from each data point were dropped from the dataset due to not being distinct enough to be useful for training, and also in an attempt to speed up training time. 66% of the remaining data was used as training data for the Naive Bayes algorithm, which trained for 20 minutes, with

the remaining 33% used as testing data after the training was complete.

7. Results and Discussion

One of the important things to note when this data is used for a Naive Bayes algorithm is that the separation of features is distinct for almost every individual feature, which allows for extremely accurate predictions with new data. These graphs depict a number of random features from the dataset. In fig. 6 the red is zipped data points and the blue is encrypted data points.

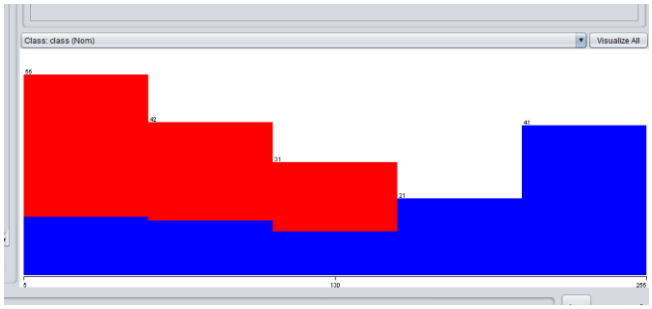

Fig. 6. Zipped data points and encrypted data points

The important point to note here is that the zipped data is always clustered together while encrypted data is usually much more spread out.

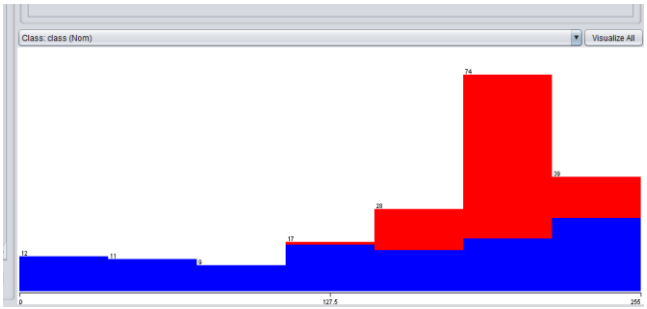

Fig. 7. zipped data points and encrypted data points

When the algorithm compares new data to features like this, it checks to see if the features in the new data point land within the cluster of data for compressed files for each individual feature. You may think that this would have a room for error if an encrypted piece of data consistently had features that fell within those boundaries, but with 40,000 individual features with this type of clustering that anomaly becomes very rare.

```
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     zip
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     rc4
Weighted Avg.    1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000

=== Confusion Matrix ===

  a  b   <-- classified as
 36  0 |  a = zip
  0 29 |  b = rc4
```

Fig. 8. Accuracy by class and confusion matrix

As shown in fig. 8when the 33% of the data set saved for tested was used on the algorithm, it had a 100% success rate in distinguishing between the two data type. Showing that using machine learning to identify between the two is entirely possible.

7.1 Improvements moving forward

In order to improve upon this algorithm, a larger dataset with more types of compression and encryption could be used in order to create an algorithm that could be useful in network activity monitoring.

8. Future Works

In future works, a large variety of data could be encrypted and compressed using multiple compression and encryption algorithms, as well as using a larger amount of data points to insure accuracy with datasets that are more random,

which could then be used to check for network activity software and digital forensics.

9. Conclusion

In conclusion using machine learning algorithms to distinguish between encrypted and compressed data is entirely possible and with a larger and more diverse dataset these algorithms can, with enough access to resources, likely be trained to the point that they can detect multiple types of encryption and compression.

10. Appendix

10.1 Bash Script to collect hexdumps of rc4 files

```
#!/bin/bash
for entry in `ls rc4Files`; do
hexdump -s0x100 -e '16/1 "%02x " "\n"' rc4Files/$entry| tr '\r\n' ' '| sed -e 's/^/rc4 /' - | sed -e 's/ /,/g' | sed -e 's/,*$//g' >/tmp/bee && echo ','>/tmp/bees && cat /tmp/bee /tmp/bees >>rc4dump.txt
done
```

10.2 Bash Script to collect hexdumps of zip files

```
#!/bin/bash
for entry in `ls zipFiles`; do
hexdump -s0x100 -e '16/1 "%02x " "\n"' zipFiles/$entry| tr '\r\n' ' '| sed -e 's/^/zip /' - | sed -e 's/ /,/g' | sed -e 's/,*$//g' > /tmp/farts && echo ','>/tmp/butts && cat /tmp/farts /tmp/butts >>zipdump.txt
done
```

10.3 Python arff converting program

```
#!/usr/bin/env python3
MAX_FEATURES = 1026718//2 + 1
with open("bigDump.txt", "r") as fp:
    contents = fp.read();
    rows = contents.split("\n");
    final = ""; # The final CSV file

    things_worked_out_ok = False
    largest = 0
    i = 0
    #figures out largest file size
    for r in rows:
        n  =r.count(",")
        if n > largest:
            largest = n
    MAX_FEATURES = largest
    #appends extra zeros to all rows
    for r in rows:
        i+=1
        number_of_cells = r.count(",")
        ifnumber_of_cells< MAX_FEATURES:
            r+="00,"*(MAX_FEATURES - number_of_cells-1)
            r+="00"
        elifnumber_of_cells == MAX_FEATURES:
            r = r[:-1]
            things_worked_out_ok = True
        else:
            print("Failure")
            exit();
        hex = r.split(",")
        t=0
        #converts hex to decimal for weka to use
        for h in hex:
            if h=="*":
                h="0"
            if h!= "rc4" and h != "zip":
                hex[t] = str(int(h,16))
            t+=1
        s=","
        c=s.join(hex)
        final += c+"\n"

    assert(things_worked_out_ok)
    assert(MAX_FEATURES > 80)
    # Make header
    header="@RELATION compressEncrypt\n@ATTRIBUTE class {zip,rc4}\n"
    fori in range(MAX_FEATURES-2):
        header += "@ATTRIBUTE "+"BEE"+str(i)+" NUMERIC\n"
    header +="@DATA\n"
    final  =header+final
    final = final[:-1]
    with open("bigDump_mod.arff", "w") as nfp:
        nfp.write(final)
```

REFERENCES

[1] K. Kaur and X. Xiaojiang Du and K. Nygard, “Enhanced routing in Heterogeneous Sensor Networks”, IEEE Computation World’09, pp. 569-574, Athens, Greece, Nov. 15-20, 2009.

[2] Dan Boneh and Victor Shoup. “A Graduate Course in Applied Cryptography”. https://crypto.stanford.edu/~dabo/cryptobook/draft_0_2.pdf, 2015.

[3] Lauren Evanoff, Nicole Hatch, Gagneja K.K., “Home Network Security: Beginner vs Advanced”, ICWN, Las Vegas, USA, July 27-30, 2015.

[4] Gagneja K.K. and Nygard K., "Heuristic Clustering with Secured Routing in Heterogeneous Sensor Networks", IEEE SECON, New Orleans, USA, pages 51-58, June 24-26, 2013.

[5] Gagneja K.K., “Knowing the Ransomware and Building Defense Against it - Specific to HealthCare Institutes”, IEEE MobiSecServ, Miami, USA, pp. 1-5, Feb. 11-12, 2017.

[6] Gagneja K.K., “Secure Communication Scheme for Wireless Sensor Networks to maintain Anonymity”,

IEEE ICNC, Anaheim, California, USA, pp. 1142-1147, Feb. 16-19, 2015.

[7] Gagneja K.K., "Pairwise Post Deployment Key Management Scheme for Heterogeneous Sensor Networks", 13th IEEE WoWMoM 2012, San Francisco, California, USA, pages 1-2, June 25-28, 2012.

[8] Gagneja K.K., "Global Perspective of Security Breaches in Facebook", FECS, Las Vegas, USA, July 21-24, 2014.

[9] Gagneja K.K., "Pairwise Key Distribution Scheme for Two-Tier Sensor Networks", IEEE ICNC, Honolulu, Hawaii, USA, pp 1081-1086, Feb. 3-6, 2014.

[10] Gagneja K., Nygard K., "Energy Efficient Approach with Integrated Key Management Scheme for Wireless Sensor Networks", ACM MOBIHOC, Bangalore, India, pp 13-18, July 29, 2013.

[11] Gagneja K.K., Nygard K., "A QoS based Heuristics for Clustering in Two-Tier Sensor Networks", IEEE FedCSIS 2012, Wroclaw, Poland, pages 779-784, Sept. 9-12, 2012.

[12] K. K. Gagneja, K. E. Nygard and N. Singh, "Tabu-Voronoi Clustering Heuristics with Key Management Scheme for Heterogeneous Sensor Networks", IEEE ICUFN 2012, Phuket, Thailand, pages 46-51, July 4-6, 2012.

[13] Gagneja K.K., Nygard K., "Key Management Scheme for Routing in Clustered Heterogeneous Sensor Networks", IEEE NTMS 2012, Security Track, Istanbul, Turkey, pp. 1-5, 7-10 May 2012.

[14] Runia Max, Gagneja K.K., "Raspberry Pi Webserver", ESA, Las Vegas, USA, July 27-30, 2015.

[15] Gagneja K., James L., "Computational Security and the Economics of Password Hacking", Future Network Systems and Security. FNSS 2017. Communications in Computer and Information Science, vol. 759. Springer. 2017.

[16] GeeksforGeeks, "RSA Algorithm in Cryptography", https://www.geeksforgeeks.org/rsa-algorithm-cryptography.

[17] A. S. Gagneja and K. K. Gagneja, "Incident Response through Behavioral Science: An Industrial Approach," 2015 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, 2015, pp. 36-41.

[18] Tirado E., Turpin B., Beltz C., Roshon P., Judge R., Gagneja K., "A New Distributed Brute-Force Password Cracking Technique", Future Network Systems and Security, FNSS Communications in Computer and Information Science, vol. 878, pp 117-127, 2018

[19] Caleb Riggs, Tanner Douglas and KanwalGagneja, "Image Mapping through Metadata," Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC), Shanghai, China, 2018, pp. 1-8.

[20] Keely Hill, Gagneja K.K., "Concept network design for a young Mars science station and Trans-planetary communication", IEEE MobiSecServ 2018, Miami, FL, USA, Feb. 24-25, 2018.

[21] Javier Campos, Slater Colteryahn, GagnejaKanwal, "IPv6 transmission over BLE Using Raspberry PI 3", International Conference on Computing, Networking and Communications, Wireless Networks (ICNC'18 WN), March 2018, pp. 200-204.

[22] Gagneja K., Jaimes L.G., "Computational Security and the Economics of Password Hacking", Future Network Systems and Security. FNSS 2017. Communications in Computer and Information Science, vol. 759, pp. 30-40, Springer, 2017.

[23] Gagneja K.K. Ranganathan P., Boughosn S., Loree P. and Nygard K., "Limiting Transmit Power of Antennas in Heterogeneous Sensor Networks", IEEE EIT2012, IUPUI Indianapolis, IN, USA, pages 1-4, May 6-8, 2012.

[24] Mehdi Chehel,Amirani, Mohsen Toorani, and Ali Beheshti, "A New Approach to Content-based File Type Detection", IEEE, 2008

[25] Sarah Simpson. "Cryptography Defined/Brief Hiestory", http://www.laits.utexas.edu/~anorman/BUS.FOR/course.mat/SSim/history.html, 1997.

[26] Sharma Ruchita and SwarnalataBollavarapu. "Data Security using Compression and Cryptography Techniques". Vol 117 - No. 14. International Journal of Computer Applications (0975-8887), 2015.

[27] William Stallings. "Cryptography and Network Security: Principles and Practice". 6th edition. Pearson Education, 2014.

[28] PKWARE Inc., ".ZIP File Format Specification", version 6.3.4, PKWARE Inc., 2014.