**detect by machine learning whether an attack using ransomware has occurred**

**Author: Orel Rahum  
Supervisors: Dr. Amit Dvir, Harel Berger**

**Introduction**

Since the 50s, the world has seen the merits and the wonders of the Internet. Internet users generate about 2.5 quintillion bytes of data each day. (and every each day its grow-up)

It is easy to see that as the size of the information that goes through the internet is greater, the need to prevent attacks over the Internet increases with it.

one kind of malware that use the importance of the internet called ransomware.

ransomware is a type of malware that prevents users from accessing their system or personal files and demands ransom payment in order to regain access.

It is a relatively new malware but has generated much interest from cybercriminals because of its successful attack and direct financial interest. Ransomware objective is to block its victim from accessing their own resources by locking the OS or encrypting targeted files that seem valuable to the victim, such as images, spreadsheets and presentations. [2]. Basically, there are two types of ransomwares: locky and crypto. Locky ransomware locks the entire system from access by its user, but it is usually easy to resolve. However, crypto ransomware uses encryption technology to lock selected files from user access; this is much more difficult to resolve and the damage caused may be irreversible. Crypto ransomware is also the more popular type employed by cybercriminals. A third type of ransomware called scareware has been mentioned in the literature [3]. This ransomware does not actually damage the victim’s computer but only scares the victim into

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

**Related work**

In this section we review existing work for detect lock files, we noticed that we could divide the used techniques for detect ransomware few categories:

1. Detection by the hardware

2. Detection by the dedicated software

3. Detection by the internet traffic

most the solutions based on Use of probability ratio according to different parameters

in the article **2entFOX A Framework for High Survivable**

article they used Bayesian network that containing common scenarios.

example of common scenario is:

1.Access to cryptographic libraries

2.Access to specific registry paths

3.Targeted files search key words

The default of the software is to take 20 different cases and after calculating by using a Bayesian network we get a probability number between 0 and 1 and we can know in high probability if we attacked or no

in the article **Detecting crypto‑ransomware in IoT networks based on energy**

The article builds on our knowledge of each device's power consumption after installing popular apps by using a learning machine.

In this article we will build a learning machine that works on a large Stock of Data-set that containing benign phone data and Encryption phones data and thus the model we can knows based on the energy if the device went through a ransomware

in the article **CryptoLock (and Drop It) Stopping Ransomware Attacks on User Data**

Cryptolock software based on Indicators (Indicator is a variable that can be only 2 value , 0 or 1)

The software is based on 3 main indicators-

1. Change the file type - These signatures describe the order

And location of specific byte values ​​unique to the file type,

Such data should be considered as suspicious.

2. Copy source file and comparison- When copying it should be similar to probability 0 ,if not - its suspect

3. Shannon Entropy - Use of the Shannon indicator that updates whether there is any uncertainty about the files

**Cryptography**

The word cryptography consists of two words. Crypto which means secret and graphics which means spelling.

Cryptography is a field thousands of years old and to some extent today's cryptography is like

The cryptography of yesteryear.

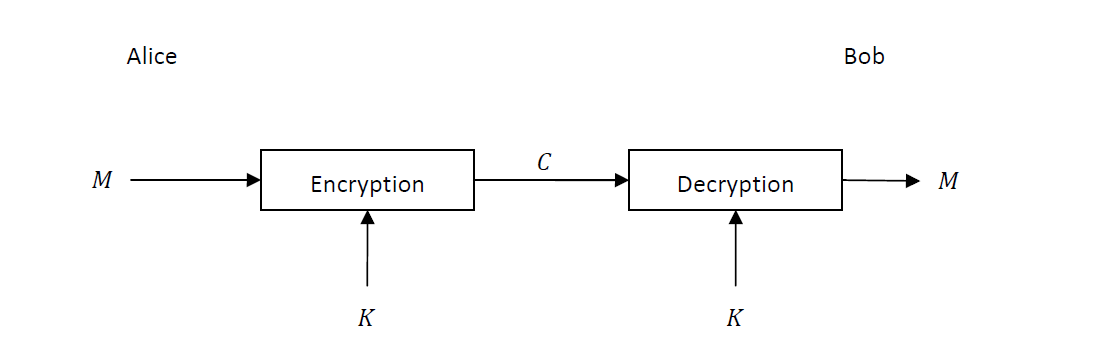
As in any field, there are people who take advantage of the industry and there are people who make malicious use of it, which we will talk about in the article.

In this article we will focus on the field of encryption in cryptography and explain the number of types of encryption and the differences between them.

**Symmetrical encryption:**

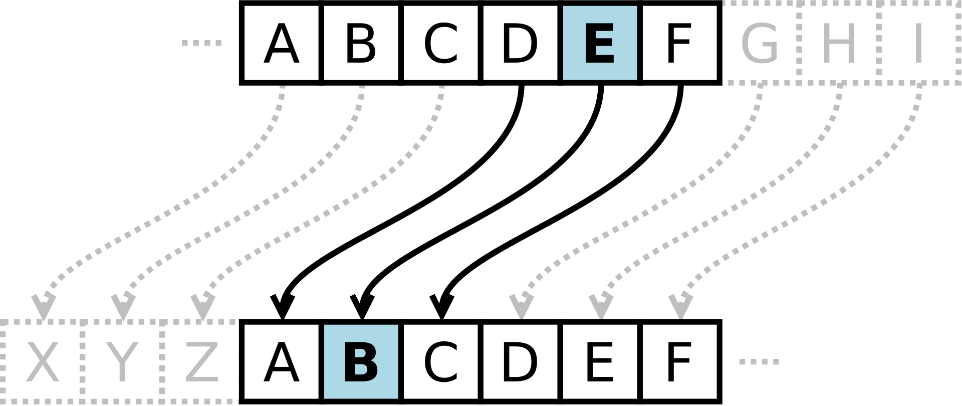
In the world of symmetric encryption, the assumption is that two people who want to communicate encrypted have a key

With which they can encrypt messages and decrypt encrypted messages or verify messages

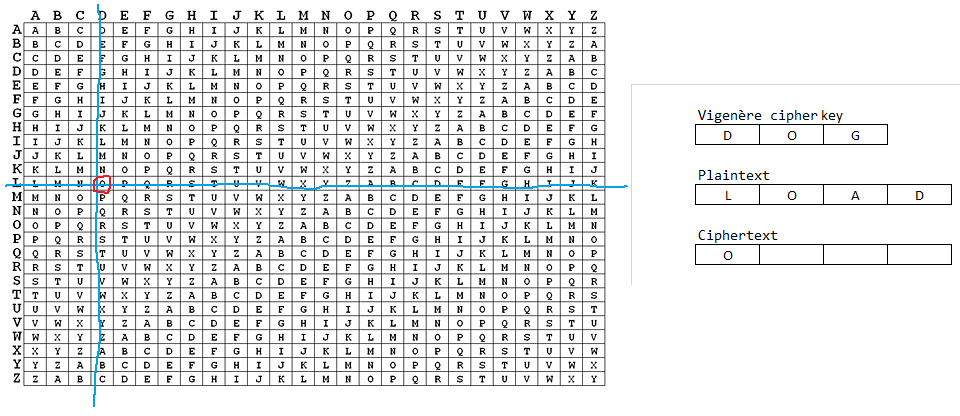


Caesar cipher

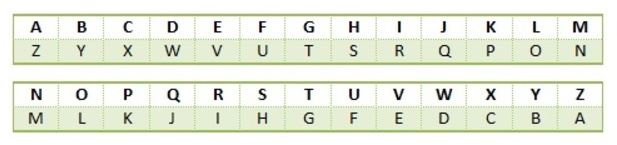
In cryptography, a Caesar cipher, also known as Caesar's cipher, the shift cipher, Caesar's code or Caesar shift, is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a left shift of 3, D would be replaced by A, E would become B, and so on. The method is named after Julius Caesar, who used it in his private correspondence.



**Vigenère cipher**

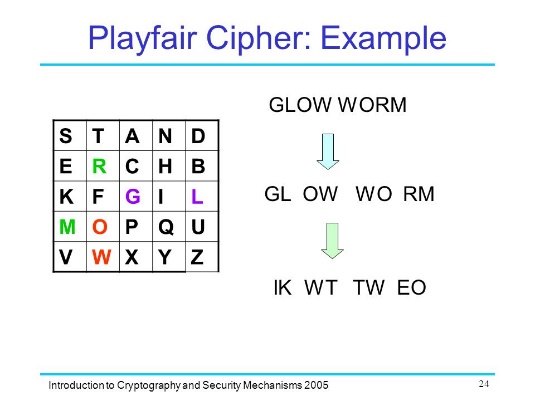
Like Caesar cipher ,which replaces each letter in a message with a different letter.  
The addition in The Visner cipher is the replaces each letter in a message with a different letter from a different alphabet, i in a different key. The use of the key is done cyclically. After using all the alphabets go back to the first alphabet. The position of each letter in the original message determines which alphabet from the alphabet group of the encryption key is encrypted. [2] In each alphabet key, the order of the letters is different, so that each identical letter in the message will be encrypted to a different letter in the cipher, so the frequency of the letters in the original message is not preserved, in contrast to a single alphabetic cipher that moves all letters at a fixed distance. Therefore, the Visner cipher cannot be cracked with the help of frequency analysis.

**Atbash cipher**

**Atbash is a monoalphabetic**[**substitution cipher**](https://en.wikipedia.org/wiki/Substitution_cipher)**originally used to**[**encrypt**](https://en.wikipedia.org/wiki/Encrypt)**the**[**Hebrew alphabet**](https://en.wikipedia.org/wiki/Hebrew_alphabet)**. It can be modified for use with any known**[**writing system**](https://en.wikipedia.org/wiki/Writing_system)**with a standard**[**collating order**](https://en.wikipedia.org/wiki/Collation).

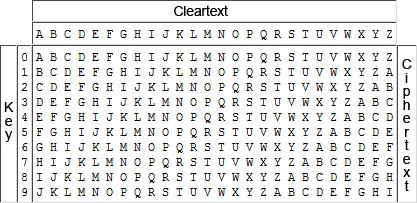
**Playfair cipher**

The technique encrypts pairs of letters ([*bigrams*](https://en.wikipedia.org/wiki/Bigram) or *digrams*), instead of single letters as in the simple [substitution cipher](https://en.wikipedia.org/wiki/Substitution_cipher) and rather more complex [Vigenère cipher](https://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher" \o "Vigenère cipher) systems then in use. The Playfair is thus significantly harder to break since the [frequency analysis](https://en.wikipedia.org/wiki/Frequency_analysis) used for simple substitution ciphers does not work with it. The frequency analysis of bigrams is possible, but considerably more difficult. With 600[[1]](https://en.wikipedia.org/wiki/Playfair_cipher#cite_note-1) possible bigrams rather than the 26 possible monograms (single symbols, usually letters in this context), a considerably larger cipher text is required in order to be useful.



**Gronsfeld Cipher**

The Gronsfeld cipher is a variant created by Count Gronsfeld (Josse Maximilaan van [Gronsveld](https://en.wikipedia.org/wiki/Gronsveld" \o "Gronsveld) né van Bronckhorst); it is identical to the Vigenère cipher except that it uses just 10 different cipher alphabets, corresponding to the digits 0 to 9). A Gronsfeld key of 0123 is the same as a Vigenere key of ABCD. The Gronsfeld cipher is strengthened because its key is not a word, but it is weakened because it has just 10 cipher alphabets. It is Gronsfeld's cipher that became widely used throughout Germany and Europe, despite its weaknesses.



**Asymmetric encryption:**In the asymmetrical world each entity has a pair of keys - a public key and a private key. The public key is used

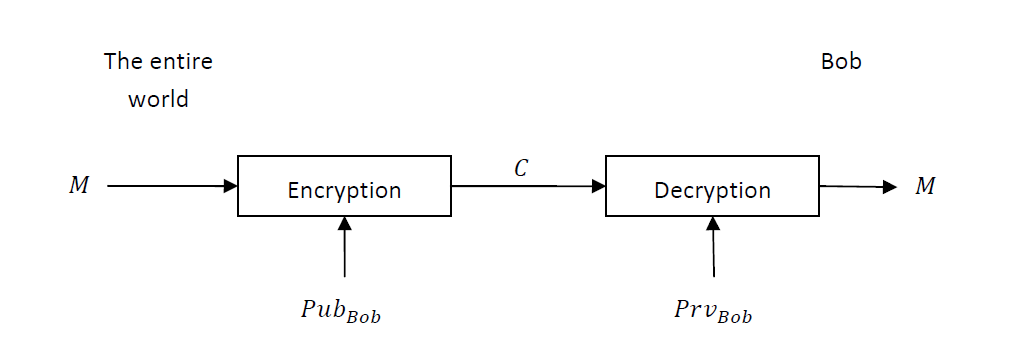
The rest of the world to send messages and verify received messages while the private key is used to decrypt

Messaging and signing outgoing messages.

This solves the problem of the number of keys in the case of symmetric cryptography. Here man has only

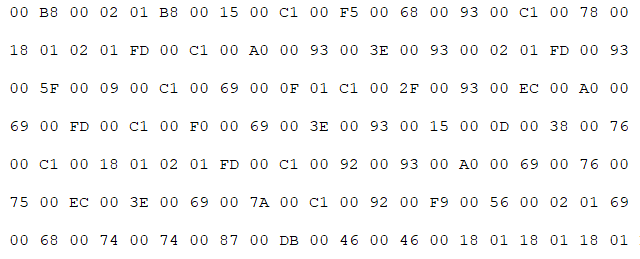
Two keys and the whole world can send him encrypted messages and he can send signed messages and anyone

Can verify the signature.

On the other hand, it is still unclear how the public keys are distributed.

**RSA Cipher**

In RSA, as with any public key system, the encryption key is not secret and is different from the secret key that is kept secret, so it is called asymmetric. The asymmetry in RSA stems from the practical difficulty of decomposing into factors of a freak number that is a multiple of two large primes, which is an open problem in number theory.



**Machine learning**

**Machine learning** (**ML**) is the study of computer algorithms that improve automatically through experience.[[1]](https://en.wikipedia.org/wiki/Machine_learning#cite_note-1) It is seen as a subset of [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence). Machine learning algorithms build a model based on sample data, known as "[training data](https://en.wikipedia.org/wiki/Training_data)", in order to make predictions or decisions without being explicitly programmed to do so.[[2]](https://en.wikipedia.org/wiki/Machine_learning#cite_note-2) Machine learning algorithms are used in a wide variety of applications, such as [email filtering](https://en.wikipedia.org/wiki/Email_filtering) and [computer vision](https://en.wikipedia.org/wiki/Computer_vision), where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

To solve the employment problem (or similar problems), we build a machine learning model (called a classifier) that, when given a new instance (a user), predicts a label (whether she is employed or not).

**Classifier:** A classifier is a special case of a hypothesis (nowadays, often learned by a machine learning algorithm). A classifier is a hypothesis or discrete-valued function that is used to assign (categorical) class labels to particular data points.

A **data set** (or **dataset**) is a collection of [data](https://en.wikipedia.org/wiki/Data). In the case of tabular data, a data set corresponds to one or more [database tables](https://en.wikipedia.org/wiki/Table_(database)), where every [column](https://en.wikipedia.org/wiki/Column_(database)) of a table represents a particular variable, and each [row](https://en.wikipedia.org/wiki/Row_(database)) corresponds to a given record of the data set in question.

Train / Test Split

* In order to train our classifier, we must have some labeled data (for example, we must know for some users whether they are employed or not).
* In order to evaluate the performance of our classifier, we split our data into train and test sets, train our classifier on the train set, and then test it on the test set.

|  |  |  |
| --- | --- | --- |
| **Confusion Matrix** | **Classified as Positive** | **Classified as Negative** |
| **Really Positive** | True Positive | False Negative |
| **Really Negative** | False Positive | True Negative |

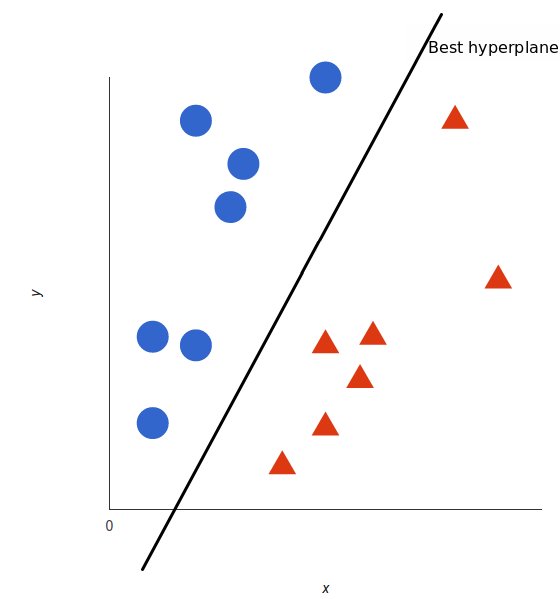
* Accuracy is how much Accuracy have my Machine learning

To calculate Accuracy need do : Trues / All or (True Positive + True Negative) / (True Positive + True Negative + False Negative + False Positive)

* Recall its mean What fraction of positives did we actually find?   
  to calculate recall need do : True Positive / Really Positive or True Positive / (True Positive + False Negative)
* Precision its mean If we say positive, how precise are we?  
  to calculate Precision need do : True Positive / Classified as Positive or True Positive / (True Positive + False Positive)

**SVM Machine learning**

A support vector machine (SVM) is a supervised [machine learning](http://www.monkeylearn.com/machine-learning/) model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they’re able to categorize new text.

****As is customary in this field, training examples are represented as vectors in linear space. For classification problems, in the training phase appropriate classifier that separates as positively as possible between positive and negative training examples. The classification created in SVM is the linear separator which creates as large a space as possible between it and the examples closest to it in the two categories. When a new point is displayed, the algorithm will detect whether it is located within the line defining the group, or outside it.

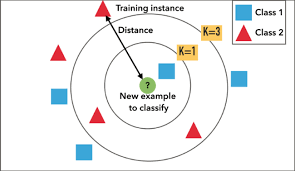
**KNN**

The Near Neighbor Algorithm (or k-NN for short) is a parameterless parameter for local classification and regression. [1] In both cases the input depends on the k closest observations in the feature space. k-NN can be used for classification or regression:

K-NN for classification - Given the input of a new example, the algorithm belongs to the group. The example is associated with the most common class among k nearest neighbors (where k is defined as an integer, usually a small number). If k = 1 the object is associated with the class of the nearest single neighbor.

K-NN for regression - Given a new example, the algorithm returns a sample property value. This value is the average of the values ​​of the k values ​​of the nearest neighbors.

k-NN is an instance-based learning algorithm, or lazy learning, where the function is only locally approximated and all calculations are postponed until it is classified. The k-NN algorithm is one of the simplest algorithms in machine learning.

****

**My DATASET**

In order to teach the machine properly, we need a large database to teach the machine.

So in order to get the best identification system we trained our machine on hundreds of thousands of files.

The information was always divided into unencrypted files and encrypted files.

It is important to note that all the unencrypted information was a lot of real books and text files

In addition, all the encrypted information was encrypted in different percentages in order to teach the machine properly and be able to reach an online solution. That you can save the file as quickly as possible

In order to achieve maximum results in the machine, all the information is divided into 3 types.

1. Only letters of English letters

2. Only characters and numbers.

3. All types of Basque characters.

Every type of text files have 50% text file that Unencrypted and 50% text file that encrypted by 6 type of encrypted : : Atbash , Autokey , Caesar , Gronsfeld , Playfair , RSA.  
My Unencrypted took from :  
1. download from this site: <http://www.gutenberg.org/> .  
2. divide few big text files to a lot   
of small text file (200-2200 rows)

For learn my ML   
All my DATASET was Translated to CSV file with all Features results.   
I Try to have a lot of type of text files with different data and different chars to get max result.

**My Features**

1. Count the number of word in my text file.  
   To know the Length of my text file I need some information to show me.   
   Count my word on my text is the best way to check it
2. Count if word start on small char  
   Because I work on on different text files, the file can have 20 row or 1200 row.
3. Count if its spam word!  
   After read about cryptographic encrypted I learn how need look Suspicious words
4. Count how much popular word in text  
   I took the popular 100 English word to array.  
   And count how much popular word in text .   
   (this feature based on feature 5 what we will explain next slide)
5. Check if the word is in English or no.  
   Basically in Descriptive word this feature is enough
6. Count how much line on text :  
   This feature is to check how much line in the text

**Final results**

Having understood a little of the field of cryptocurrency and a little of the field of machine learning.

I can explain to you my solution method.

After an in-depth investigation of hundreds of thousands of text files, I saw quite a few patterns that were repeated in the encryption.

Most and all of the encryptions come in one case to a word that cannot be pronounced at all and in the other case to a word that does not exist in the dictionary.

The ideal solution is to go over all the words in the text, and if it is a word that exists in the dictionary keep it as a fitz.

This solution is recommended for offline work on the files. But in order to test all the words in hundreds of thousands of files, it takes quite a few weeks for the machine to work to reach the output.

To find the balance between as accurate answers as possible, even with partial encryption and the speed of the solution.

After a lot of experiments I did, I found the combination of efficiency and correctness.

1. Count the number of word in my text file.  
   To know the Length of my text file I need some information to show me.   
   Count my word on my text is the best way to check it
2. Count if word start on small char  
   Because I work on on different text files, the file can have 20 row or 1200 row.
3. Count if its spam word!  
   After read about cryptographic encrypted I learn how need look Suspicious words
4. Count how much popular word in text  
   I took the popular 100 English word to array.  
   And count how much popular word in text .   
   (this feature based on feature 5 what we will explain next slide)
5. Check if the word is in English or no.  
   Basically in Descriptive word this feature is enough
6. Count how much line on text :  
   This feature is to check how much line in the text.

My result is :

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Precision** | **Recall** | **F1-Score** |
| SVM | 91% | 91% | 91% |
| KNN | 92% | 91% | 91% |

