International Baccalaureate

Extended Essay

To what extent is the Advanced Encryption Standard information-theoretically secure?

Name: Awin Gray

Subject: Computer Science

Candidate Number: 003411-0027

Session: May 2016

School: St. Andrews International School Bangkok

School Code:

Supervisor: Andrew Bramwell

Total Word Count: 3,772

ABSTRACT

This paper evaluates the Advanced Encryption Standard when subjected to changes in key sizes and input sizes, in order to see how secure the encryption standard is in relation to key sizes and input sizes, and what implications are given by the results obtained.

In order to understand the relation and implications, this investigation uses perfect secrecy as a measurement of security. The research gives a brief insight into the topic of cryptography and information theory, it first covers the history and development of encryption standards, then the higher level design of the current encryption standard is described, and finally perfect secrecy is introduced as a definition of security. A test with varied input sizes were conducted in a controlled manner, obtained results are recorded and presented in a tabular form in order to visualize the correlation between input sizes and secrecy values, along with the distinction between key sizes and secrecy values.

It was concluded that the Advanced Encryption Standard is indeed secure in terms of perfect secrecy, but it does not have perfect security. The results showed consistency within the algorithm and it was indicated that changing the input size does not affect the security of algorithm to a great extent. On the other hand, the test carried out did not take in other factors that are normally present in real-life usage, therefore the results obtained do not represent practical scenarios but rather give a theoretical overview of how secure the algorithm is.

Word Count: 245

Table of Contents

1. Introduction

"It was thanks to ULTRA that we won the war." (Brown, 1987, p. 671). During World War II, the British intelligence recruited allied code breakers to crack the secret code of the Enigma, a machine that encoded all of Germany's wartime communication. With 10^{16} possible settings of the Enigma, it was thought to be unbreakable. However, a flaw in the Enigma's mechanism was discovered, which subsequently led to the designation of ULTRA, a secret project dedicated to decoding all of secret communication of the Axis powers (Lendl, 2012). The breaking of Enigma was considered by Supreme Allied Commander Dwight D. Eisenhower to have made a significant contribution to the Allied victory.

Alan Turing, the leading British cryptanalyst who devised techniques for cracking coded messages, also developed an electromechanical machine that could derive the settings for Enigma which was estimated to shorten the time of war by two to four years. During that time, Turing was assigned to share his methods of breaking coded messages in Washington where he met an American mathematician and electronic engineer, Claude Shannon (Hodges, 1999). Over a cup of tea, Turing introduced the idea of "Universal Turing Machine", an abstract theory of computational machine, to Shannon (Hodges, 1999). As the war comes to an end, Shannon and his colleagues published a paper on signal processing and data smoothing which commenced the information age and digital revolution (Mindell, 2002). Shortly after the war, "A Mathematical Theory of Communication" appeared in the 1948 July and October issues of the Bell System Technical Journal. Shannon defines the concept of information and how it can be quantified using probability theory, information entropy is introduced as a measure of uncertainty in a message, and essentially his concepts formed the field of information theory (Shannon & Weaver, 1949).

Inspired from wartime research, Shannon published "Communication Theory of Secrecy System" where he addresses an interesting application of cryptography in his theory of communication. One of the most important concept in Shannon's theory of secrecy system is the measure of secrecy which can serve as a metric for evaluating the security of a cryptosystem, Shannon defines two notions for security: Information-Theoretical Security and Computational Security, both can also be identified as theoretical and practical secrecy. The notions of security devised by Shannon greatly influenced the development of modern cryptography, notably public-key cryptography which it was discovered that shared key between communication parties is not necessary for secrecy (Golomb, et al., 2002).

The rapidly increasing number of communications systems and the introduction of the internet has brought high demand for security services and measures to protect digital information, over the past decade the internet has experienced an 806.0% growth in the number of users connected to the internet worldwide (Internet World Stats, 2015). The Advanced Encryption Standard, also known as the AES, is an accepted encryption scheme used by the US government to protect classified information (National Institute of Standards and Technology, 2001). Although it became a federal government standard, it was also made publicly available for use in protecting non-classified information. The AES adopts an algorithm called Rijndael, the selection process of the algorithm has ensured the best possible encryption scheme for the standard. It provides fast encryption (Schneier & Whiting, 2000), making it suitable for software applications and especially implementation in firmware or hardware such as routers or firewalls. Modern applications also employ security protocols such as SSL or TLS, which rely on AES for their encryption functionality to ensure secure transmission of information over the World Wide Web (McGrew & Bailey, 2015).

From winning the war to protecting your credentials, the field of cryptography has had a significant impact on communication. The Enigma was once thought to be impossible to break, but eventually it was proven wrong and the following consequences were devastating. I ask myself "Would it ever happen again even with today's modern cryptography?" the question has led me to investigate the extent of which the AES is information-theoretically secure. Using Shansnon's definition of information-theoretical security to investigate modern cryptosystem like the AES, will provide an insight on how secure it really is, the outcome of this extended essay should justify how important encryption is to us.

2. History and Early Development of the Encryption Standard

In the early 1970s, non-military research about cryptographic algorithms was nearly inexistent and not a lot of people understood the field of cryptography. In 1972, the former US "National Bureau of Standards" (NBS) initiated a program with the goal of protecting sensitive and unclassified government data (Tuchman, 1998), one of their aims included a development of a single standardized cryptographic algorithm, such that it could be tested and certified, and different equipment using algorithm could interoperate easily.[1]

A year later on, the National Bureau of Standards issued a public request for proposals of the standardized algorithm, the request has raised public interest thus the request has exposed the idea

[1] Interoperability refers to the ability for different information systems to communicate and exchange data.

of encryption to the general population, however there were very little expertise, since none of the proposed algorithms barely met the requirements defined by the National Bureau of Standards (Tuchman, 1998). A second request was issued one year after the first, and the National Bureau of Standards eventually received a promising candidate, and it was an algorithm called Lucifer which had been developed by International Business Machines Corporation (IBM). The National Bureau of Standards requested help of the National Security Agency (NSA), and opened the discussion to the general public and the cryptographic community. The Lucifer algorithm was then later modified to become the Data Encryption Standard (DES), which it was claimed to be free of any mathematical or statistical weaknesses (Schneier, 1996).

After its reign, Data Encryption Standard was put through multiple tests and it was first publicly broken during a challenge called DESCHALL proposed by RSA Security in 1997. A group of computer scientists have gain access to DES-encrypted message through the use of brute force key search, with computational resources offered by volunteers through a distributed network (Curtin & Dolske, 1998).

Within the same year Data Encryption Standard was broken, the National Institute of Standards and Technology (NIST) announced that they wished to replace Data Encryption Standard (DES) as a specification for electronic data encryption. In 2001, the Advanced Encryption Standard (AES) was announced by NIST as a replacement cipher. Following the announcement, a five-year-long selection process in which fifteen proposed designs for the new specification were analyzed and evaluated extensively by the cryptographic community worldwide. (National Institute of Standards and Technology, 2001) The algorithm which was selected as the Advanced Encryption Standard was called Rijndael, it was proposed by two Belgian cryptographers Vincent Rijmen and Joan Daemon. Other finalists in the selection process included RC6, Serpent, MARS, and Twofish.

Schneier and Whiting (2000) have done extensive investigations on the performance of Rijndael compared to other finalists in the selection process, and their findings have shown Rijndael to be more efficient in both encryption and decryption process.

Meanwhile, an effective comparison of AES and DES has been conducted, it was examined that the structure and designs of Rijndael algorithm such as the "non-linearity of key expansion" which eliminates the possibility of equivalent keys, providing AES a significant advantage over DES and

the Rijndael cipher has been considered as an unbreakable algorithm even in modern day computing (N. Penchalaiah et al., 2010).

3. Design and Implementation

Rijndael is a suite of cryptographic algorithms with different key and block sizes. For the Advanced Encryption Standard, the National Institute of Standards and Technology selected three members of the Rijndael ciphers, each member consists of the same fixed block size of 128 bits but with different key sizes ranging from 128 bits, 192 bits and a maximum of 256 bits. By contrast, the Rijndael specification actually consists of block and key sizes that may be any multiple of 32 bits, both with a minimum of 128 and a maximum of 256 bits. (Daemen & Vincent, 2002)

The Rijndael cipher used in Advanced Encryption Standard is a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the plain text data. Rijndael is also a block cipher, essentially a cryptosystem which consists of functions that map n-bit of plain text blocks into n-bit of cipher text blocks, and their main objective is to provide confidentiality.

The design of Rijndael algorithm is based on the idea of processing a 128-bit plain text data and mapping it onto a $4 * 4$ matrix of bytes ordered as shown in **Figure 3.1.**

$$\begin{pmatrix} \text{byte0} & \text{byte4} & \text{byte8} & \text{byte12} \\ \text{byte1} & \text{byte5} & \text{byte9} & \text{byte13} \\ \text{byte2} & \text{byte6} & \text{byte10} & \text{byte14} \\ \text{byte3} & \text{byte7} & \text{byte11} & \text{byte15} \end{pmatrix}$$

Figure 3.1. Diagram illustrating a four by four matrix of bytes

Although some members of Rijndael ciphers have a larger block size and have additional columns in the matrix, most of the calculations are done in a special finite field.[2] The first step in the encryption process is called key expansions, the round keys are derived from Rijndael's key scheduling algorithm. The initial round of the encryption process involves XOR operation on each plain text input data with the corresponding byte of the first round key. Then a fixed number of rounds based on a substitution-permutation network is applied.[3]

[2] See Daemen and Vincent pages 10 to 15 for a more detailed description of finite fields.

[3] See Daemen and Vincent pages 19 to 22 for a more detailed description of substitution-permutation operations.

The encryption process is made up of 3 different number of rounds of processing: 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each rounds of processing are identical, except for the last rounds of each encryption process. The key size of the algorithm specifies the number of repetitions of transformation rounds.

Each round consists of four sequences of transformation called steps, including one single-byte based substitution step called $SubByte$, a circular shift row-wise permutation step called $ShiftRows$ applied to all rows in the matrix, a linear transformation column-wise mixing step called $MixColumns$ applied to each column, and the XOR operation of the round key called $AddRoundKey$.

The decryption process of the algorithm consists of reversing the round steps while using the same key as one used in the encryption process, these reverse operations are applied to a cipher text to retrieve the plain text.

4. Communication Theory of Secrecy System

As the main objective of block ciphers such as Rijndael is to provide confidentiality, a corresponding objective of an adversary is to be able to recover plain text data from cipher text without knowing the key used in the encryption process. The block cipher is said to be totally broken if the adversary can obtain information about the key, and partially broken if they can obtain parts of plain text from the cipher text. When evaluating the security of a cryptosystem, certain assumptions have to be made. The first assumption that should always be made is that an adversary has access to all data transmitted over a communication channel, we also use Kerchkhoff's principle which assumes that an adversary has knowledge of the cryptosystem and its encryption function. The principle has been reformulated by Claude Shannon as "the enemy knows the system". (Shannon, 1949)

The metrics used in the security evaluation of cryptosystems have been devised by Shannon, the first model is called computational security. It is a security model that concerns the computational effort required to break a cryptosystem. A cryptosystem is computationally secure if the algorithm used to break the cryptosystem requires at least N operations. However, this definition of security is more

likely to be applicable to certain types of attacks that require more computation (e.g. exhaustive key search attack).[4]

Another security notion formulated by Claude Shannon was called Provable Security, this is a different approach which attempts to measure or prove a cryptosystem by the means of reduction. In simpler terms, this definition of security attempts to show that if a cryptosystem can be broken in a specific way, it also would be possible to efficiently solve problems that are thought to be difficult; this is similar to proving that certain problems are NP-Complete or solving difficult problems such as integer factorization.

Finally there is unconditional security, a security model that assumes infinite computation allowed for an adversary, an unconditionally secure cryptosystem means that it cannot be broken given infinite computational resources. The term unconditional security is also used interchangeably with information-theoretic security, for encryption schemes, unconditional security is called perfect secrecy.

For a symmetric-key encryption scheme to be have perfect secrecy, the key must be at least the same length as cipher text. The adversary should not be able to determine the key given cipher text, the plain text and any symbol or information about the plain text. As we allow infinite computational resources, the appropriate framework for measuring unconditional security or perfect secrecy is probability theory.

5. Shannon's Idea of an Encryption Scheme

Mathematically, a cryptosystem can be defined as a tuple (P, C, K, E, D) where P is a set of possible plain texts called plain text space, C is a set of possible cipher texts called cipher text space, and K is a set called key space and its elements are called keys. E $= \{E_k : k \in K\}$ is a set of functions $E_k : P \rightarrow C$. Its elements are called encryption functions. Finally, D $= \{D_k : k \in K\}$ is a set of functions $D_k : C \rightarrow P$ and its elements are called decryption functions.

[4] Data Encryption Standard has been shown to be computationally unsecure, see Curtin and Dolske for more details on brute force key search attack

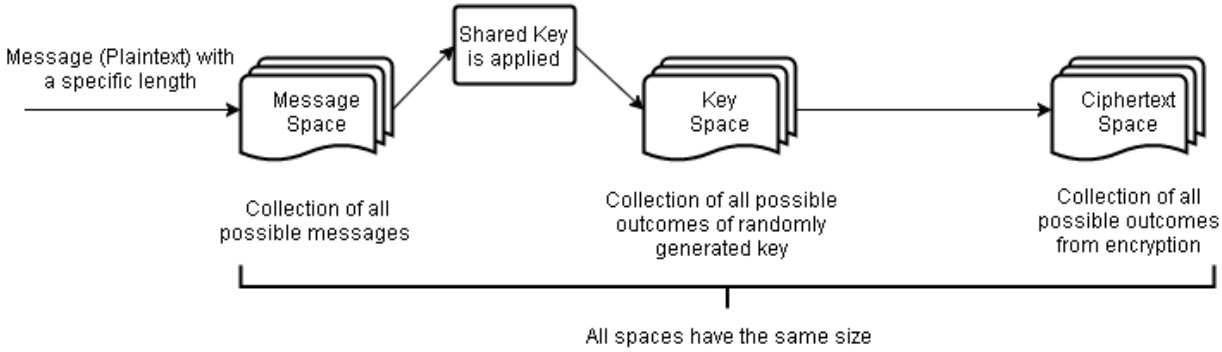

Figure 4.1 An illustration of encryption scheme described by Shannon

If we suppose there's a probability distribution on a message space $\{P_1, P_2, \dots, P_n\}$, and the key space $\{K_1, K_2, \dots, K_n\}$ are also distributed with known probabilities. By Shannon's original definition, a cryptosystem is perfectly secure if $\Pr[x|y] = \Pr[x]$ for all $x \in P, y \in C$. What the formula means is that: the a-posteriori probability that the plain text is x, given that the cipher text y is intercepted, is identical to the a-priori probability that the plain text is x. In simpler terms, observing an intercepted cipher text does not give the adversary any more information about plain text which the adversary does not already know from the a-priori message distribution of plain text space. For an adversary who only has access to a page from the cipher text space, the only knowledge they have is that all messages in the message space is equally likely. Even with unlimited computational power, an adversary is only able to guess which renders high computational power useless. A possible flaw we can identify from the scheme is the generation of shared key, the adversary may be able to obtain information about the key through randomness analysis where the pattern of key generation can be determined if the key generator is not truly random.

The following definition of perfect secrecy is restricted to a scenario where a key is used for only one encryption of the plain text, in order to realistically measure the security of a cryptosystem, the idea of entropy derived by Shannon is used in a situation where more plain texts are encrypted using the same key.[5] Assume that X is a discrete random variable from a finite set X. The entropy of the message X is defined to be $H(X)$ as:

$$H(X) = -\sum_{x \in X} \Pr[x] log_2 \Pr[x]$$

The entropy $H(X)$ is the least quantity of bits required to encode all possible meanings of the message X, assuming the probability of all messages occurring $\Pr[x]$ are equally likely.

[5] Cryptanalyst can carry out a ciphertext-only attack when the same key is used to encrypt more plain texts.

The entropy of a message can also tell us the uncertainty, where the uncertainty of a message is the quantity of plain text bits that an adversary must recover from the given cipher text in order to obtain the plain text. Lastly, secrecy value of a cryptosystem is calculated in terms of key equivocation denoted as $H_c(K)$, where key equivocation of a key K given cipher text C is the conditional entropy of plain text P given cipher text C, i.e. key equivocation is the uncertainty of a message which is reduced when additional information is provided. Finally we get our secrecy value formula:

$$H_c(K) = \sum\{C\}\, P(C) \sum\{K\} P_c(K) log_2[P_c(K)]$$

6. Testing

The investigation of this research aims to evaluate Advanced Encryption Standard using Perfect Secrecy as a security model. The testing factors include varying key sizes from 128 bits to 256 bits, varying plain text sizes from 10 Kilobytes to 100 Kilobytes and a single mode of operation to encrypt these data. A cryptographic module called "PyCrypto" written by Dwayne Litzenberger was used to implement the AES cipher and a scientific computing module called "NumPy" was used to manipulate data and perform mathematical operations.[6]

The method of assessing secrecy begins with a calculation of byte distribution within the key. A function called $countByteDist(data)$ takes in an array of byte values as a parameter. The algorithm works by iterating through a byte array and recording how often each byte appears by constructing an array called $countedData$ with a size of 256 cells imitating the size of a byte i.e. 8-bit value. Given a cipher C, the algorithm computes the probability $P_c(K)$ of each byte in the cipher text that appears in the key K, then the total sum of $P_c(K) log_2 P_c(K)$ gives the entropy of the key K given cipher C.

The second part of the algorithm attempts to find $P(C)$ by using the same function $countByteDist(data)$ to calculate how often each cipher byte has appeared in cipher text C. The probability of each byte appearing in the key K is then computed and summed all together for all possibilities of the cipher bytes. This cipher text is obtained after the plain text operations with the key; i.e. this cipher text is correlated to the above key.

[6] Visit https://github.com/dlitz/pycrypto for the source code repository

The final step in finding secrecy of a cipher is the process of summing the product of the entropy of key K given cipher C and $P(C)$, the summation of all possibilities return the secrecy value $H_c(K)$. (Weerasinghe, 2014)

Advanced Encryption Standard will be tested under 128-bit, 192-bit and 256-bit keys initialized with random bytes using a cryptographically stronger version of Python's standard "random" module. All encryption done in this research will be operating in Electronic Code Book (ECB) mode, because it is the simplest encryption mode where each plain text blocks is directly encrypted into a cipher text block. Conducting the experiment in only ECB mode makes the encryption independent of any other blocks, creating a controlled environment which ensures the size of plain text as the only independent variable. Other assumptions in this investigation are that the key is shared and that the cryptosystem encrypts a single block of message.

The cipher algorithm will be tested with large variation of input data sizes ranging from a minimum of 10 Kilobytes to a maximum of 100 Kilobytes. A plain text is initialized with the same module to create random bytes of a specific size, then the encryption algorithm returns a cipher text as a byte string. After the cipher text output is obtained, a function called "fromstring" from a scientific computing module "NumPy" is used to convert cipher text and the corresponding key from byte string to a one-dimensional array of unsigned 8 bits integer representing the cipher text and its key used in the encryption, the array is passed as input data to calculate secrecy value.

The algorithm is repeated 100 times to obtain an average secrecy value for each data size, this is done to ensure reliability of the results obtained. The results will be visualized in a graph where the relationship between data size and average secrecy value is demonstrated along with their corresponding key size.

7. Results and Evaluation

7.1 Presenting results

	Average Secrecy Values		
Input Size (kB)	128-bit key	192-bit key	256-bit key
10	0.1320	0.2243	0.2879
20	0.1327	0.2238	0.2726
30	0.1323	0.2241	0.2747
40	0.1325	0.2238	0.2695
50	0.1330	0.2130	0.2914
60	0.1326	0.2155	0.2763
70	0.1327	0.2242	0.2937
80	0.1327	0.2062	0.3035
90	0.1327	0.2080	0.2839
100	0.1327	0.2000	0.3141

Table 6.1. Tabular results of the large variation test of data sizes

Table 6.1. shows no continuous correlation between input sizes and secrecy values, the difference in secrecy values for each key sizes clearly indicate that the higher the size of the key, the higher the secrecy value meaning that they are more secure. The secrecy values obtained in each input sizes and key sizes indicate no correlation, in fact rather random but within the order of magnitude of each corresponding key sizes. 128 bits key is in the order of 0.1 of the secrecy values, 192 bits key is in the order 0.2 and 256 bits key is close to the order of 0.3 of the secrecy values.

7.2 Evaluation

The results clearly show that the Advanced Encryption Standard is very consistent in the encryption process with a small range of secrecy values for each key sizes. Change in input sizes do not seem to make a significant impact on the secrecy of the cipher, this means that the size of plain text data does not affect the amount of possible information about the key making Advanced Encryption Standard a cryptosystem that is less prone to attacks that require computational efforts, this is because the secrecy values are roughly the same despite plain text data sizes.

The highest secrecy value obtain is 0.3141 using a 256-bit key, the value shows that the Advanced Encryption Standard does not conform to the definition of perfect secrecy meaning some information about the key can be extracted from the plain text data, however this does not mean that Advanced Encryption Standard is not highly secure. The results obtained from this investigation should be compared with other known encryption standard especially the Data Encryption Standard. This investigation left out many factors such as the operation modes of the block cipher, the type of data to be encrypted, or the speed of the encryption/decryption algorithm, therefore the results obtained are not realistic. Repeating the calculation and computing the average secrecy value gives the results reliability and better accuracy as a single calculation of secrecy value can give a range of approximately up to 1%.

8. Conclusion

The results obtained during the test shows slight variation of secrecy values between input sizes, the difference in secrecy between key sizes are highly noticeable as expected. The consistency in secrecy values indicate a strong cryptographic algorithm.

Because the only testing factors were input sizes and key sizes, the scope of this investigation is limited to effectively assess the whole algorithm, however the investigation rather provides a general overview of the security provided by the Advanced Encryption Standard.

Overall, the consistency of secrecy values throughout various input sizes and the clear positive correlation between secrecy and key sizes make Advanced Encryption Standard rather secure in terms of perfect security despite seemingly negative results. The unrealistic scenario of this investigation does not equate impractical results, the results from this investigation show that Shannon's definitions of security are likely to be too strong to be realistic in practice.

Works Cited

Brown, A. C., 1987. *C.* 1st ed. New York: Macmillan.

Curtin, M. & Dolske, J., 1998. A Brute Force Search of DES Keyspace. *USENIX.*

Daemen, J. & Vincent, R., 2002. *The Design of Rijndael.* s.l.:Springer.

Golomb, S. W. et al., 2002. Claude Elwood Shannon (1916-2001). *Notices of the American Mathematical Society,* 49(1), pp. 8-16.

Hodges, A., 1999. *Turing.* 1st ed. New York: Routledge.

Internet World Stats, 2015. *World Internet Users Statistics and 2015 World Population Stats.* [Online]
Available at: http://www.internetworldstats.com/stats.htm
[Accessed 11 October 2015].

Lendl, C., 2012. *Bletchley Park: British Cryptanalysis during World War II,* Austria: s.n.

McGrew, D. & Bailey, D. V., 2015. *RFC 6655 - AES-CCM Cipher Suites for Transport Layer Security (TLS).* [Online]
Available at: https://tools.ietf.org/html/rfc6655
[Accessed 11 October 2015].

Mindell, D. A., 2002. *Between Human and Machine.* 1st ed. Baltimore: John Hopkins University Press.

N. Penchalaiah et al., 2010. Effective Comparison and Evaluation of DES and Rijndael algorithm (AES). *International Journal on Computer Science and Engineering (IJCSE),* pp. 1641-1645.

National Institute of Standards and Technology, 2001. *Announcing the Advanced Encryption Standard (AES),* s.l.: s.n.

Schneier, B., 1996. *Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C.* s.l.:John Wiley & Sons, Inc..

Schneier, B. & Whiting, D., 2000. A Performance Comparison of Five AES Finalists. *AES Candidate Conference,* pp. 123-135.

Shannon, C., 1949. Communication Theory of Secrecy Systems. *The Bell System Technical Journal,* pp. 656-715.

Shannon, C. E. & Weaver, W., 1949. *A Mathematical Theory of Communication.* 1st ed. Urbana: University of Illinois Press.

Tuchman, W., 1998. A brief history of the data encryption standard. In: *Internet Besieged.* New York: ACM Press/Addison-Wesley Publishing Co., pp. 275-280.

Weerasinghe, T., 2014. A Tool to Analyse Symmetric Key Algorithms. *International Journal of Information & Network Security (IJINS),* pp. 26-32.

Appendix A: Source code for secrecy value calculation

```
from Crypto.Cipher import AES
from Crypto import Random
import numpy as np
import math

def countByte(data):
    countedData = [0] * 256
    for k in data:
        countedData[k] += 1
    return countedData

def calculateSecrecy(key, cipher):
    countedKey = countByte(key)
    countedCipher = countByte(cipher)

    entropy = 0
    secrecy = 0

    for j in range(0, 256):
        p_k = 1.0 * countedKey[j] / len(key)
        p_c = 1.0 * countedCipher[j] / len(cipher)
        if (p_k > 0):
          entropy += p_k * np.log2(p_k)
          secrecy += -p_c * entropy

    return secrecy

def encryptAES(key, plaintext):
    cipher = AES.new(key)
    ciphertxt = cipher.encrypt(plaintext)
    return ciphertxt

def getResults(keysize, plaintextsize):
    key = Random.new().read(keysize)
    totalValue = 0

    for i in range(0, 100):
        plaintext = Random.new().read(plaintextsize)
        ciphertxt = encryptAES(key, plaintext)

        cipherbyte = np.fromstring(ciphertxt, dtype=np.uint8)
        keybyte = np.fromstring(key, dtype=np.uint8)
        totalValue += calculateSecrecy(keybyte, cipherbyte)

    avgValue = totalValue/100
    return avgValue
```