International Journal of Information & Network Security (IJINS)
Vol.3, No.1, February 2014, pp. 26~32
ISSN: 2089-3299

A Tool to Analyze Symmetric Key Algorithms

T.D.B Weerasinghe
MSc.Eng, BSc.Eng(Hons), C|EH, MIEEE, AMIE(SL), AMCS(SL)
Software Engineer – IFS R&D International (Pvt) Ltd, Sri Lanka.

Article Info

Article history:

Received Nov 12th, 2013
Revised Dec 20th, 2013
Accepted Jan 26th, 2014

Keywords:

Symmetric key ciphers
Cryptogrpahic tools
Secrecy of ciphers

ABSTRACT

With the growth of the internet and the interconnectivity of computer and data networks, security of data transmission has always been a concern of many stakeholders of information and communication arena. Among them, the researchers who work in the field of Cryptography and Network Security pay a lot attention to deliver highly secured and cost effective security mechanism and/or systems. Many types of cryptographic tools are available in open literature/internet, but this tool provides a mechanism to visualize the security levels of the symmetric key algorithms w.r.t Shannon's theories on secrecy of ciphers. The tool can be used to analyze the secrecy and performance levels of many symmetric key algorithms and it is capable of analyzing plaintext in the form of character inputs (passwords). More importantly this tool and can be extended to evaluate combined algorithms as well as new symmetric key algorithms, hence the targeted users of this tools are researches and software engineers who are in the field of Cryptography and Network Security

Corresponding Author:

T.D.B Weerasinghe
MSc.Eng, BSc.Eng(Hons), MIEEE, AMIE(SL), AMCS(SL),
Software Engineer – IFS R&D International (Pvt) Ltd, Sri Lanka.
Email: tharindu.weerasinghe@gmail.com

1. INTRODUCTION

The tool introduced by this paper, will be helpful to researchers, software engineers who work in the field of Cryptography as they need to analyze security levels of the algorithms. Especially in circumstances where they need to have numerical values to describe the secrecy. Since this tool can be extended to develop hybrid algorithms (combining symmetric key algorithms) without developing an attack on those new algorithms they will be able to analyze the secrecy and performance by using this tool. This is an outcome of some literature analysis of crypto tools available in the internet [4, 5, 6, 7, 8, and 9]. As many of them are encryption tools that gives a secure cipher-text as outputs but this tool is for research and software engineering community who works in the area of cryptography, especially in the area of symmetric key algorithms.

In open literature, when we search cryptographic tools we find many software products that help the users to get encrypted material. In other words, there are a lot of tools to make use of the existing encryption algorithms. In this research the focus was to provide a tool evaluate existing algorithms with some new algorithms (combined algorithms like AES+RC4 as well as new symmetric key algorithms)with the existing algorithms. This tool can be used by the researchers who havejava programming knowledge, i.e. to extend the tool's behavior to analyze newly created algorithms (since this open source anyone can use and edit the

code). The other idea was to give a numerical output to depict the secrecy of each cipher rather than simulating some known attacks. That reduces the complexity of the tool and makes it more sensible. If a new symmetric key algorithm is introduced by someone, then surely this tool can be used to analyze its secrecy and performance with respect to other ciphers. (For example if it is a new stream cipher or a variant of an existing one, then it can be compared with the other stream ciphers which are already in the forefront.)

Technology used to develop this tool:
Core Java
Java Cryptography Package (Javax Crypto)
Netbeans IDE

Platform:
Intel® Core™ i3 CPU, M370 @ 2.40 GHz with 1.86 GB usable RAM in Microsoft Windows 7 Home Basic (32 bit)

Special Features:

- User friendly interface.
- Can be used to combine block or stream ciphers and analyze them.
- Since Javax crypto package is used the correctness of the algorithms can be trusted.
- Since this tool is dedicated for researchers and software engineers who work in the field of Information Security, they can extend this tool to analyze the secrecy and performance of symmetric key algorithm they create!
- Not complex. Secrecy is analyzed using theoretical definitions of Shannon, but it is a good measure to get an idea about the ciphers. Higher the secrecy (numerical value) higher the security!

2. OTHER TYPES OF TOOLS AVAILABLE

Tool introduced by Bozga L et al mainly focuses on cryptographicprotocols and their implementation. So, it does not purely targetalgorithms but set of protocols like Schroeder-Loweprotocol. And also it focuses on secrecy not the performance [1]. Tool introduced by Blanchet also focuses on protocols andverifying their authenticity. In this tool, pi calculus is used torepresent each protocol with some fairly random cryptographic primitives [2]. Tools described in the reference [3] which is freely available online, aresome e-Learning tools related to Cryptography. They can bevery useful to learn about Cryptographic algorithms.Although they have many features, there is a lack of focustowards secrecy analysis using Shannon's theories as well asthe performance of the algorithms and most of them arelimited for Windows operating system. On the other handthere are limitations when it comes to develop newalgorithms and analyze them.Thus it is obvious that each tool is different and has differentobjectives. Tools described above are pretty good tools thatcan be used to various purposes. The tool presented in thispaper mainly focuses on a simple way to depict the secrecyand performance of symmetric key ciphers and this tool ismade for people who have programming knowledge so thatthey can extend the tool for new algorithms. Secrecy andperformance calculations are written in separate classes sothat those can be easily used to analyze new algorithms.

3. TOOL DESCRIBED IN THIS PAPER

This tool consists of two parts.

1. Calculation of Secrecy: Depends of theories of secrecy of ciphers by Shann
2. Calculation of Performance

The following two figures show how the tool looks like in run time:

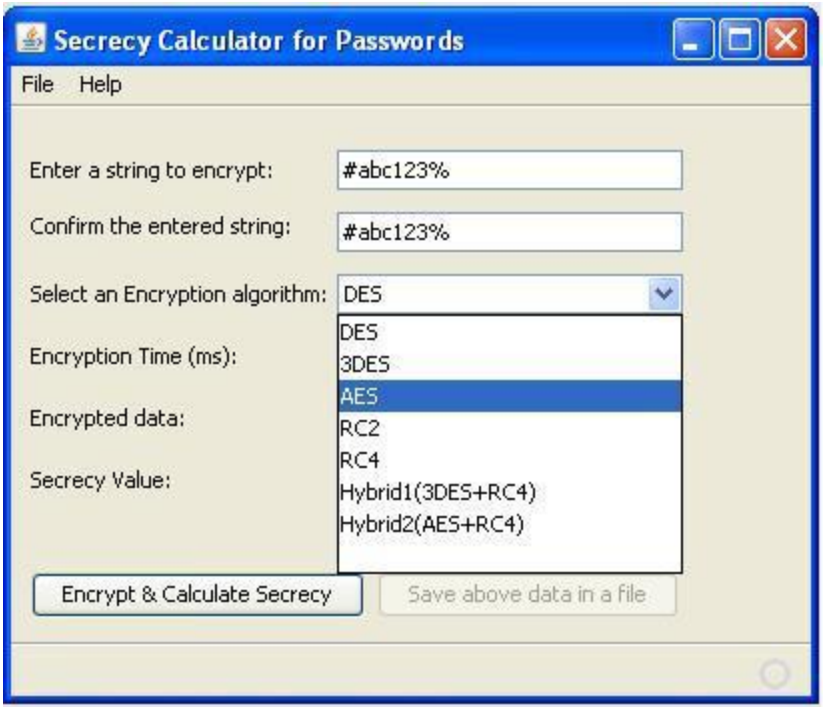

Fig.1. Algorithm List

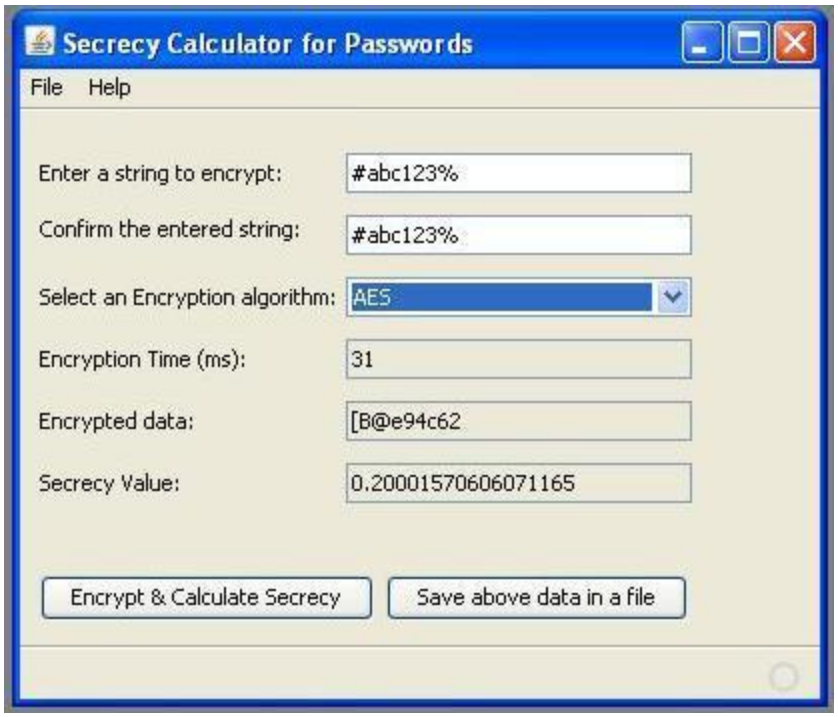

Fig.2. A Sample Output

a. SHANNON'S SECRECY OF CIPHERS

Secrecy of a cipher: Secrecy of a cipher is described in terms of the key equivocation, H (K) c of a key K for a given cipher text C; that is the amount of uncertainty in K given C. [Equivocation is the uncertainty of a message, reduced when there is additional information; Uncertainty of a message is the number of plaintext bits that must be recovered when the message is encrypted, in-order to obtain the plaintext. The uncertainty of a message is measured by its entropy. Higher the number, higher the uncertainty; Entropy of a message X is called H(X), which is the minimum

number of bits needed to encode all possible meanings of the message assuming the occurrences of all messages are equally likely.] [8]

Mathematical Equation:

$$H_c(K) = \sum\{C\}P(C)\underbrace{\sum\{K\}P_c(K)\log_2[P_c(K)]}_{\textbf{\textit{Part 1}}}$$

All the above definitions and equations are illustrated from the lecture notes of Dr.Issa Traore on Shannon's secrecy, University of Victoria, British Columbia, Canada, which were available online.

b. METHOD OF CALCULATING SECRECY IN THIS RESEARCH

- Consider the ***Part 1*** first: It is the entropy of K given the relevant cipher. (Cipher text C, has been obtained using this particular key K)
  - Calculate how often each key byte is appeared in the key.
  - And then calculate the probability of each byte appears (given the cipher) in the key and get the summation of Pc(K) * $\log_2$Pc(K).
- After that consider the other part: Calculating P(C) and then the summation.
  - Calculate how often each cipher byte has appeared in the cipher text.
  - And then calculate the probability of each byte appeared in the key and get the summation (for all possibilities of the cipher bytes). This cipher is obtained after the plaintext operations with the key; i.e. this cipher is correlated to the above key.
  - Then get the multiplication of "***Part 1***" and P(C) is calculated and finally the summation of all possibilities is calculated.

Important: Higher the value, higher the secrecy. i.e. The cipher is better!

c. IMPLEMENTATION OF SECRECY CALCULATION IN JAVA

```
public class SecrecyCalculator
{
        private static int[]

        countByteDistribution(byte[] data, int start, int length)
        {
                final int[] countedData = new
                int[256];
                for (int i=start; i<start+length; i++)
                {
                        countedData[data[i] & 0xFF]++;
                }
                return countedData;
        }

        private static double log2(double d)
        {
                return Math.log(d)/Math.log(2.0);
        }

        public static double calculateEntropy(byte[]data, int start, int length)
        {
                double entropy = 0;
                final int[] countedData = countByteDistribution(data, start, length);

                for (int i=0;i<256;i++)
                {
                        final double p_k = 1.0 * countedData[i] / length;
                        if (p_k > 0)
                        {
```

```
                                entropy += -p_k * log2(p_k);
                        }
                }
                return entropy;
        }

        public static double calculateSecrecy(byte[] key, byte[] cipher, int start)
        {
                double entropy = 0;
                double secrecy = 0;
                System.out.println("\n\t\tKey Length: " + key.length);
                final int[] countedKey = countByteDistribution(key, start,
        key.length-1);
                final int[] countedCipher = countByteDistribution(cipher, start,
        cipher.length-1);

                for (int i=0;i<256;i++)
                {
                        final double p_k = 1.0 *
                        countedKey[i] / key.length;
                        final double p_c = 1.0 *
                        countedCipher[i] / cipher.length;

                        if (p_k > 0)
                        {
                                entropy += p_k * log2(p_k);
                                secrecy += -p_c * entropy;
                        }
                }
                return secrecy;
        }
}
```

4. VERIFICATION AND ANALYSIS OF THE TOOL

Average secrecy and performance analysis were performed using the tool. Similar numbers of experiments/tests were considered for all circumstances to calculate the average values in-order to obtained reasonable outputs. Input: A password which has alphanumeric characters with special characters, which meant to be a strong password. (Example: #abc123%)

Table 1. Algorithm Vs Average Secrecy

Algorithm	***Average Secrecy***
DES	0.211263
3DES	0.255173
AES	0.268375
RC2	0.165872
RC4	0.140538
3DES+RC4	0.286776
AES+RC4	0.32096

Table 2. Algorithm Vs Average Encryption Time

Algorithm	***Average Encryption Time***
DES	25
3DES	22
AES	24.8
RC2	15.4
RC4	9.4
3DES+RC4	25.2
AES+RC4	40

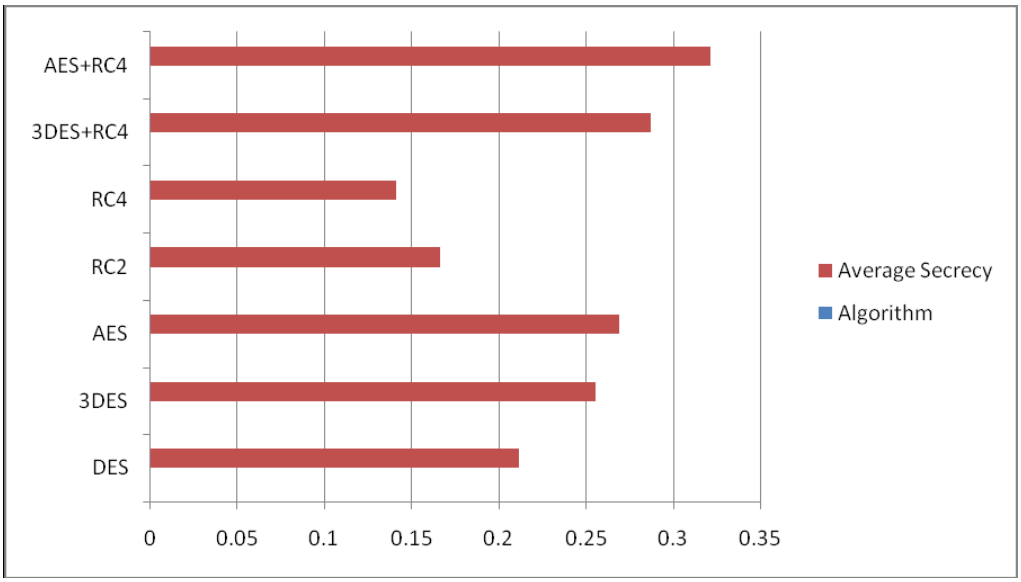

Fig. 3 Algorithm used in the tools Vs Secrecy value calculated by the tool for the sample input

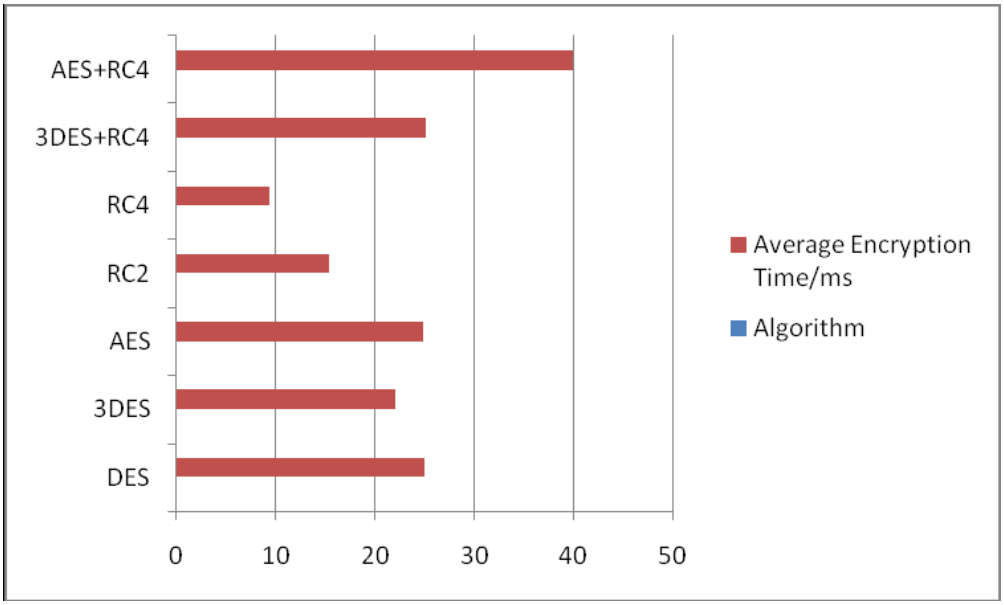

Fig. 4 Algorithm used in the tools Vs Encryption Time (ms) calculated by the tool for the sample input

4.1 RELIABILITY OF THE RESULTS OF TOOL:

With respect to Secrecy: Since the experiments are carried-out for all known and well established block and stream ciphers and according to the average secrecy values obtained the algorithms can be sorted in the descending order as follows:

AES>3DES>DES>RC2>RC4

This result is acceptable as this kind is proved by many researches before (i.e block ciphers are highly secured than stream ciphers and AES it the best block cipher around and also 3DES also commonly used.)

Combination of block and stream ciphers would give higher secrecy and as expected the tool has given the results.

Hence the results obtained from the tool are reliable w.r.t Secrecy. With respect to Performance: It is a known and proven fact that the block ciphers are complex than stream ciphers hence they are expensive than stream ciphers. The encryption times obtained by the tool also prove it. If the algorithms are sorted according to the *cost effectiveness* we have the following pattern: RC4>RC2>3DES>DES>AES *Note: Combination of block and stream ciphers would give lower performance and as expected the tool has given the result.* Hence the results obtained from the tool are reliable w.r.t Performance as well.

4.2 EXTENSIBILITY OF THE TOOL

As mentioned earlier, this tool can be extended if one wants to analyze a newly implemented symmetric key algorithm in Java. Although Shannon's theories are not 100% practical we can get an idea of the analysis. Source code can be published online.

5. CONCLUSION

This tool will be helpful to the users to evaluate the secrecy and throughput of ciphers. It will help the users to obtain numerical values for the secrecy of ciphers. The objective of this work is not to introduce yet another tool to deliver cipher text of plain texts according to the ciphering algorithms, but to help the users to use this as an analyzer which is fairly simple. Based on the Shannon's theories and encryption times an initial idea can be obtained if a new symmetric key algorithm is analyzed with the help of this tool.

The target users of this tool are software engineers in the felid of information security or information security analysts because this tool can be customized in order adopt to the new symmetric key algorithms.

6. REFERENCES

[1] L. Bozga, Y. Lakhnech and M. Périn, "HERMES: An Automatic Tool for Verification of Secrecy in Security Protocols", Computer Aided Verification, Lecture Notes in Computer Science, Volume 2725, 2003, pp. 219-222 [15th International Conference, CAV 2003, Boulder, CO, USA, July 8-12, 2003, Proceedings].
[2] Blanchet, "From Secrecy to Authenticity in Security Protocols", Static Analysis, Lecture Notes in Computer Science, Volume 2477, 2002, pp. 342-359 [9th International Symposium, SAS 2002 Madrid, Spain, September 17–20, 2002, Proceedings].
[3] CRYPTOOL PORTAL: http://www.cryptool.org/en/
[4] Advanced Crypto Software Collection: http://hms.isi.jhu.edu/acsc/
[5] Cipher Tools: http://rumkin.com/tools/cipher/
[6] Cryptography Tools: http://msdn.microsoft.com/en-us/library/windows/desktop/aa380259%28v=vs.85%29.aspx
[7] JavaScrypt: Browser-Based Cryptography Tools: http://www.fourmilab.ch/javascrypt/
[8] Lecture notes of Dr.Issa Traore on Shannon's theories, University of Victoria, British Columbia, Canada, which were available online.
[9] T.D.B Weerasinghe, "Analysis of a Modified RC4 Algorithm", International Journal of Computer Applications, vol. 51, no. 22, pp. 13-17, available at http://www.ijcaonline.org/archives/volume51/number22/8341- 1617

BIOGRAPHY OF AUTHOR

T.D.B WEERASINGHE
MSc.Eng, BSc.Eng(Hons), C|EH, MIEEE, AMIE(SL), AMCS(SL)
Software Engineer, IFS R&D International, 363, Udugama, Ampitiya Road, Kandy.

Contact No: 0094 716 860 396
Email: tharindu.weerasinghe@gmail.com
Postal Address (Home): 296, Kandy Road, Millawa, Kurunegala 60000, Sri Lanka.