Special Proceedings: ISBN #: 978-81-950383-1-2
24th Annual Conference, 23-27 February 2022; pp 45-55

Statistical Analysis of AEAD Ciphers for Transport Layer Security: An Experimental Approach

Jitendra Kurmi , Suresh Prasad Kannojia
Department of Computer Science, Lucknow University, Lucknow, UP, India -226007

Received: 12 May 2022; Revised: 22 June 2022; Accepted: 29 June 2022

Abstract

Nowadays, data security or information over the network is more critical as technology grows. To secure the data on network transport layer security SSL/TLS is used. TLS uses different ciphers of Authenticated Encryption and Associated Data (AEAD) to encrypt data during the transmission. In this paper, we perform statistical analysis of AEAD ciphers as AES_GCM (Galois Counter Mode), AES_SIV (Synthetic Initialization Vector), and AES_GCM_SIV with key size 128/256 bit based on encryption/decryption time for a message of block size 128/1024/2048/4096/8192 bit on Windows, Linux and Mac operating systems. We also measure the central tendency to determine where the most values fall in distribution, which will help us to identify the best suitable AEAD encryption algorithm. On the other hand, we use analysis of variance (ANOVA), the purpose of ANOVA is to measure the differences in strength of AEAD algorithms with different key sizes.

Key words: Authenticated encryption; SSL/TLS; AEAD; Security; Network.

1. Introduction

Cryptography algorithms are the building blocks of security and are widely used by many people and organizations worldwide. Encryption is used to protect the data from unauthorized access before transmitting the data by the sender, and decryption at receiver end. Cryptography already provides a lot of encryption/decryption algorithms. But most cryptographic algorithms do not provide confidentiality, integrity, and authenticity of data over the network. Authenticated Encryption (AE) is came into existence to deal with this problem. The AE and AEAD algorithms guarantee the confidentiality and integrity of data transmitted over the network. Various AEAD algorithms already exist, such as AES_GCM, AES_SIV, AES_GCM_SIV, Counter with CBC-MAC (CCM), Chacha20Poly1305, Deoxys, EAX, MGM, and Xsalsa20Poly1305. But in this paper, we are only considering three widely used AEAD algorithms for statistical analysis purpose.

This paper consist six sections as follows. Section 2 provides the related work that various research enthusiasts have done. The flow diagram represents the proposed methodology that is given in Section 3. Section 4 presents the experimental setup, and experimental result analysis has been done in Section 5. Finally, conclusion has given in Section 6.

2. Literature Review

In communication, message security by the asymmetric encryption technique guarantees both privacy and integrity, and it is considered authenticated encryption. Informally, such a

Corresponding Author: Jitendra Kurmi
Email: jitendrakurmi458@gmail.com

strategy ensures that no adversary can generate a ciphertext that decrypts to a valid plaintext, encryptions are indistinguishable from one another. Authenticated encryption was formerly accomplished using the "encrypt-then-authenticate" paradigm, which specifies that the resultant ciphertext should first be encrypted before applying a message authentication code. This strategy is sound, although it is inefficient most of the time. A more comprehensive analysis of composition approaches was carried out, taking into account a variety of alternatives and security goals. However, dedicated encryption modes optimized for high performance have been proposed in several circumstances.

Consider an encryption method that accepts a short secret key as input and creates a lengthy key stream as output, and used to encrypt the message bits by adding that is (modulo 2). The generated key stream must be 'pseudo-random, ' an essential security requirement for such a system. The distinction between true randomness and pseudo-randomness is a complicated one. On the other hand, the key stream should passes numerous well-known statistical tests for pseudo-randomness at the most fundamental level. Passing these tests is an essential but not sufficient requirement. The runs and autocorrelation tests are two well-known tests contains a more thorough list Knuth (2014). In an intriguing test has been developed that can be considered as universal bit generator by Maurer (1992).The chi-squared test is commonly used for determining a given sequence that follows a particular distribution or not. A brief discussion of this strategy may be found in Hell *et al*. (2009). Hypothesis testing is a practical statistical framework for analyzing cryptanalytic attacks. Often, an assault may be modeled as a hypothesis test to see if a parameter equals one of two possible values. Estimating the effort required for a successful attack becomes much more accessible when seen in this light. This approach provides a formal treatment Vaudenay (1996); Junod and Vaudenay (2003); Junod (2003). Various Statistical techniques have grown significantly to analyzing data from various power measurements is presented by Prouff *et al.* (2009). The AEAD algorithms are vulnerable to forgery and salamander attacks. An attack detection framework was proposed for authenticated ciphers to deal with this problem Kannojia and Kurmi (2021). A comparative analysis of various TLS libraries has been done, including authenticated encryption cipher, hashing, and public-key cryptography Kannojia and Kurmi (2021). Various TLS libraries have been analyzed based on supported languages, cryptographic token interface, thread safety, and CPU-assisted cryptography with Kannojia and Kurmi (2021).

3. Proposed Methodology

AEAD algorithms are used in TLS web servers for secure data transmission. The statistical randomness in encryption/decryption time of authenticated encryption algorithms with key sizes 128 bit and 256 bit are used to measure the central tendency. The purpose of measuring the central tendency is to determine where the most values fall in a distribution or not. In this paper, we proposed a statistical architecture to measure the central tendency of three AEAD algorithms with three operating systems such as Windows, Linux, and Mac shown in Figure 1.

To measure central tendency, we first implement these algorithms in python and measure the time taken by authenticated encryption ciphers to encrypt/decrypt a message of block size 128/1024/2048/4096/8192 bit with key sizes 128 bit and 256 bit. The time taken by AEAD ciphers is measured in milliseconds (ms), and further central tendency is calculated.

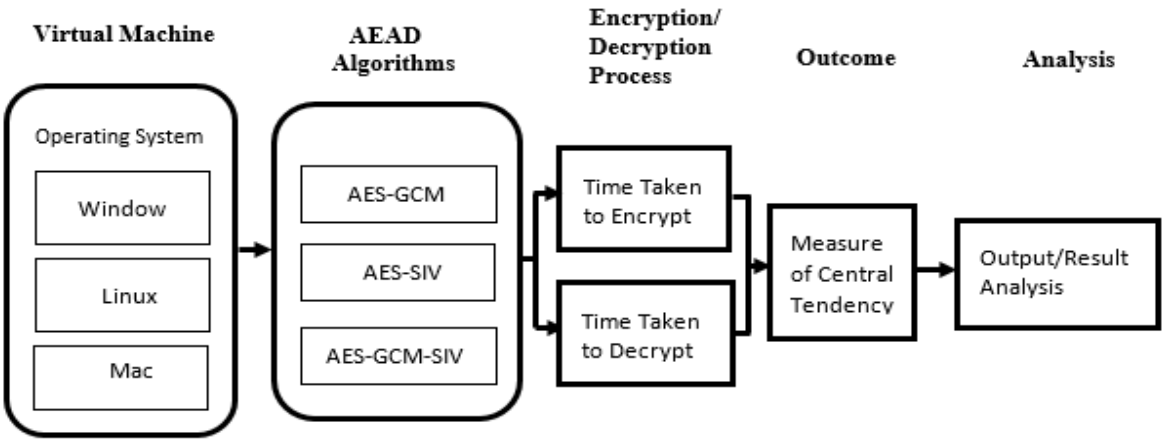

Figure 1: Proposed Architecture for Statistical analysis of AEAD ciphers

4. Experimental Setup

We performed the test on an Intel Core i3-3217U 1.80GHz CPU (4 cores) with 8GB of RAM, running Windows 10 professional machine, in a virtual environment with three operating systems Windows, Linux, and Mac.

5. Experimental Result and Analysis

5.1. Measurement of Central Tendency

The findings of the experimental results have been compiled and given in this sections, rounded upto two decimal points. The mean, standard deviation (σ_x), and standard error of the mean ($\overline{\sigma x}$) were calculated using equations (1), (2) to estimate the statistical error of the observed data (Xi) across the number of runs (N) using equation (3).

$$\overline{X} = \frac{1}{N}\sum_{i=1}^{N} Xi \tag{1}$$

$$\sigma x = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Xi - \overline{X})^2} \tag{2}$$

$$\overline{\sigma x} = \frac{1}{\sqrt{N}}\sigma_x \tag{3}$$

The measure of central tendency on windows, Linux, and Mac operating systems are shown in Tables 1, 2, and 3 and bar chart visualization method is used and presented in Figures 1 and 2.

The experimental result shown in Table 1 show that the time is taken to encrypt the message block by AEAD algorithm AES_GCM_SIV with the key size 128/256 bit is small AES_GCM_SIV 128/256 bit encryption algorithm is faster than AES_GCM and AES_SIV. The time to decrypt the message block by AEAD algorithm AES_GCM with key size 128 bit

is faster than AES_SIV and AES_GCM_SIV. However, with a 256 bit key size, the AES_GCM_SIV is faster than AES_GCM and AES_SIV on the windows operating system.

Table 1: The measure of Central Tendency on Windows Operating System

	Ciphers	Time taken to Encrypt/Decrypt Message size (ms)					Mean	Std.Dev (%)	Error (%)
		128 bit	1024 bit	2048 bit	4096 bit	8192 bit			
Windows	AES_GCM (128 bit) Encryption	255	272	314	336	354	306.20	0.4191	0.1874
	AES_SIV (128 bit) Encryption	250	267	298	313	348	295.20	0.3857	0.1725
	AES_GCM_SIV (128 bit) Encryption	252	261	282	305	339	287.80	0.3518	0.1573
	AES_GCM (128 bit) Decryption	245	261	334	356	363	311.80	0.5502	0.2461
	AES_SIV (128 bit) Decryption	251	266	345	364	379	321.00	0.5855	0.2619
	AES_GCM_SIV (128 bit) Decryption	257	272	357	373	389	329.60	0.6073	0.2716
	AES_GCM (256 bit) Encryption	391	562	687	806	960	681.20	2.1895	0.9792
	AES_SIV (256 bit) Encryption	386	554	657	799	959	671.00	2.2048	0.9860
	AES_GCM_SIV (256 bit) Encryption	385	542	638	793	957	663.00	2.2130	0.9897
	AES_GCM (256 bit) Decryption	388	552	686	807	969	680.40	2.2435	1.0033
	AES_SIV (256 bit) Decryption	381	548	677	801	958	673.00	2.2277	0.9963
	AES_GCM_SIV (256 bit) Decryption	375	541	667	791	947	664.20	2.2072	0.9871

The experimental result shown in table 2 shows that the time is taken to encrypt the massage block by AEAD algorithm AES_GCM_SIV with the key size 128/256 bit is small AES_GCM_SIV 128/256 bit encryption is faster as compared to AES_GCM and AES_SIV. The time to decrypt the message block by AEAD algorithm AES_GCM with key size 128 bit is faster than AES_SIV and AES_GCM_SIV. However, with a 256 bit key size, the AES_GCM_SIV is faster than AES_GCM and AES_SIV on Linux operating system.

Table 2: The measure of Central Tendency on Linux Operating System

	Ciphers	Time taken to Encrypt/Decrypt Message size (ms)					Mean	Std.Dev (%)	Error (%)
		128 bit	1024 bit	2048 bit	4096 bit	8192 bit			
Linux	AES_GCM (128 bit) Encryption	249	269	317	339	358	306.40	0.4618	0.2065
	AES_SIV (128 bit) Encryption	246	266	302	312	347	294.60	0.3963	0.1772
	AES_GCM_SIV (128 bit) Encryption	247	267	279	299	332	284.80	0.3244	0.1451
	AES_GCM (128 bit) Decryption	241	268	327	365	354	311.00	0.5424	0.2426
	AES_SIV (128 bit) Decryption	247	272	336	366	369	318.00	0.5565	0.2489
	AES_GCM_SIV (128 bit) Decryption	252	277	349	368	392	327.60	0.6024	0.2694
	AES_GCM (256 bit) Encryption	385	556	682	816	951	678.00	2.2041	0.9857
	AES_SIV (256 bit) Encryption	381	552	659	809	953	670.80	2.2191	0.9924
	AES_GCM_SIV (256 bit) Encryption	379	549	642	803	948	664.20	2.2074	0.9872
	AES_GCM (256 bit) Decryption	383	546	680	811	972	678.40	2.2835	1.0212
	AES_SIV (256 bit) Decryption	379	542	676	805	975	675.40	2.3033	1.0301
	AES_GCM_SIV (256 bit) Decryption	370	538	671	797	967	668.60	2.3008	1.0290

The experimental result shown in table 3 shows that the time is taken to encrypt the massage block by AEAD algorithm AES_GCM_SIV with the key size 128/256 bit is small AES_GCM_SIV 128/256 bit encryption is faster as compared to AES_GCM and AES_SIV. The time to decrypt the message block by AEAD algorithm AES_GCM with key size 128 bit is faster than AES_SIV and AES_GCM_SIV. However, with a 256 bit key size, AES_GCM_SIV is faster than AES_GCM and AES_SIV on Mac operating system.

Table 3: The measure of Central Tendency on Mac Operating System

	Ciphers	Time taken to Encrypt/Decrypt Message size (ms)					Mean	Std. Dev(%)	Error (%)
		128 bit	1024 bit	2048 bit	4096 bit	8192 bit			
Mac	AES_GCM (128 bit) Encryption	253	267	317	332	355	304.80	0.4336	0.1939
	AES_SIV (128 bit) Encryption	252	270	302	317	349	298.00	0.3833	0.1714
	AES_GCM_SIV (128 bit) Encryption	256	266	285	309	342	291.60	0.3467	0.1551
	AES_GCM (128 bit) Decryption	241	263	337	359	366	313.20	0.5741	0.2568
	AES_SIV (128 bit) Decryption	253	264	345	363	373	319.60	0.5680	0.2540
	AES_GCM_SIV (128 bit) Decryption	261	269	352	371	383	327.20	0.5792	0.2590
	AES_GCM (256 bit) Encryption	387	565	684	810	962	681.60	2.2108	0.9887
	AES_SIV (256 bit) Encryption	389	556	668	803	960	675.20	2.2001	0.9839
	AES_GCM_SIV (256 bit) Encryption	382	547	641	798	952	664.00	2.2059	0.9865
	AES_GCM (256 bit) Decryption	384	556	683	802	966	678.20	2.2342	0.9992
	AES_SIV (256 bit) Decryption	382	552	677	801	962	674.80	2.2317	0.9981
	AES_GCM_SIV (256 bit) Decryption	379	548	669	795	951	668.40	2.2031	0.9852

The error percentage Encryption/Decryption of the AEAD algorithm on the different operating systems are measured and presented in Figure 2. The error percentage of encryption of AES_GCM_SIV 128 bit and encryption of AES_GCM 256 bit is less on Windows and Linux, while the error percentage of decryption AES_GCM 128 bit and AES_GCM_SIV 256 bit is less on windows as well as Linux. The error percentage of encryption of AES_GCM_SIV 128 bit and AES_GCM_SIV 256 bit is less on Mac, while the error percentage of decryption of AES_GCM 128 bit and AES_GCM_SIV 256 bit is less on Mac.

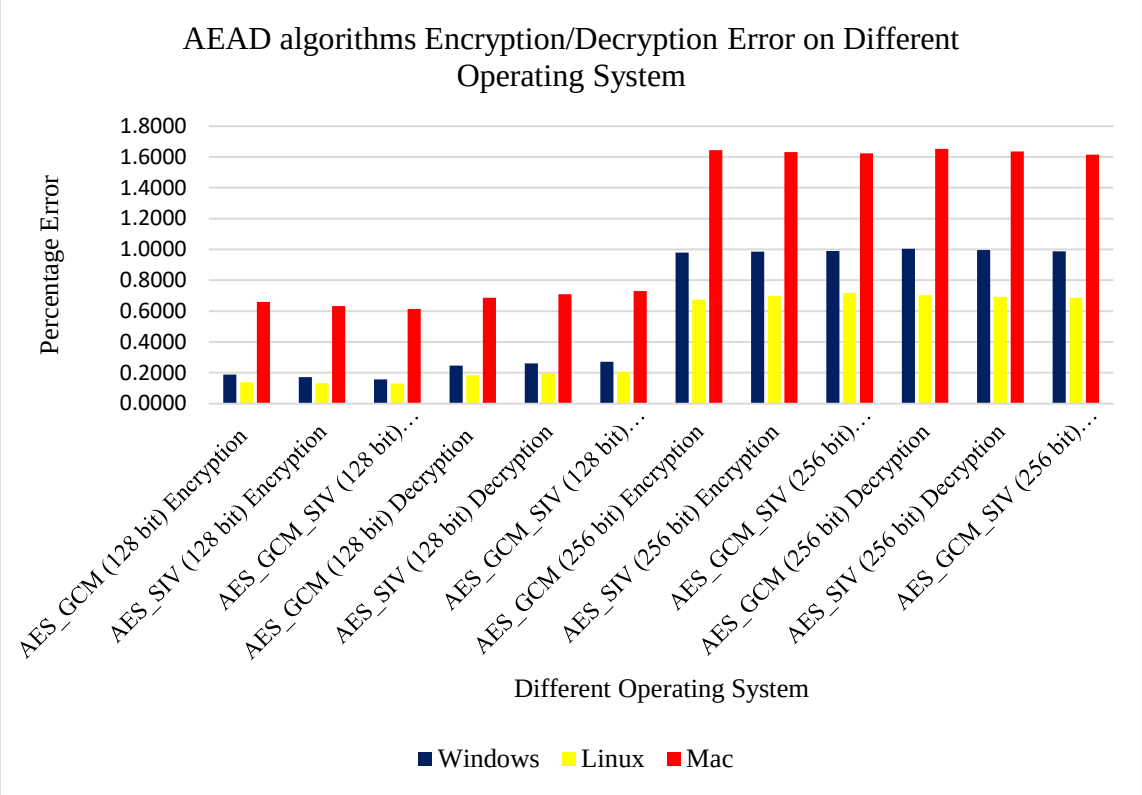

Figure 2: Comparison of AEAD algorithms Encryption/Decryption Error with three Operating System

The measure of Standard Deviation and central tendency of the AEAD algorithm on the different operating systems is measured and presented in Figure 3. The Standard Deviation and central tendency of encryption of AES_GCM_SIV 128 bit and encryption of AES_GCM 256 bit is slight on Windows and Linux, while the Standard Deviation and central tendency of decryption AES_GCM 128 bit and AES_GCM_SIV 256 bit is slight on windows as well as Linux. The Standard Deviation and central tendency of encryption of AES_GCM_SIV 128 bit and AES_GCM_SIV 256 bit is slight on Mac, while the Standard Deviation and central tendency of decryption AES_GCM 128 bit and AES_GCM_SIV 256 bit is slight on Mac.

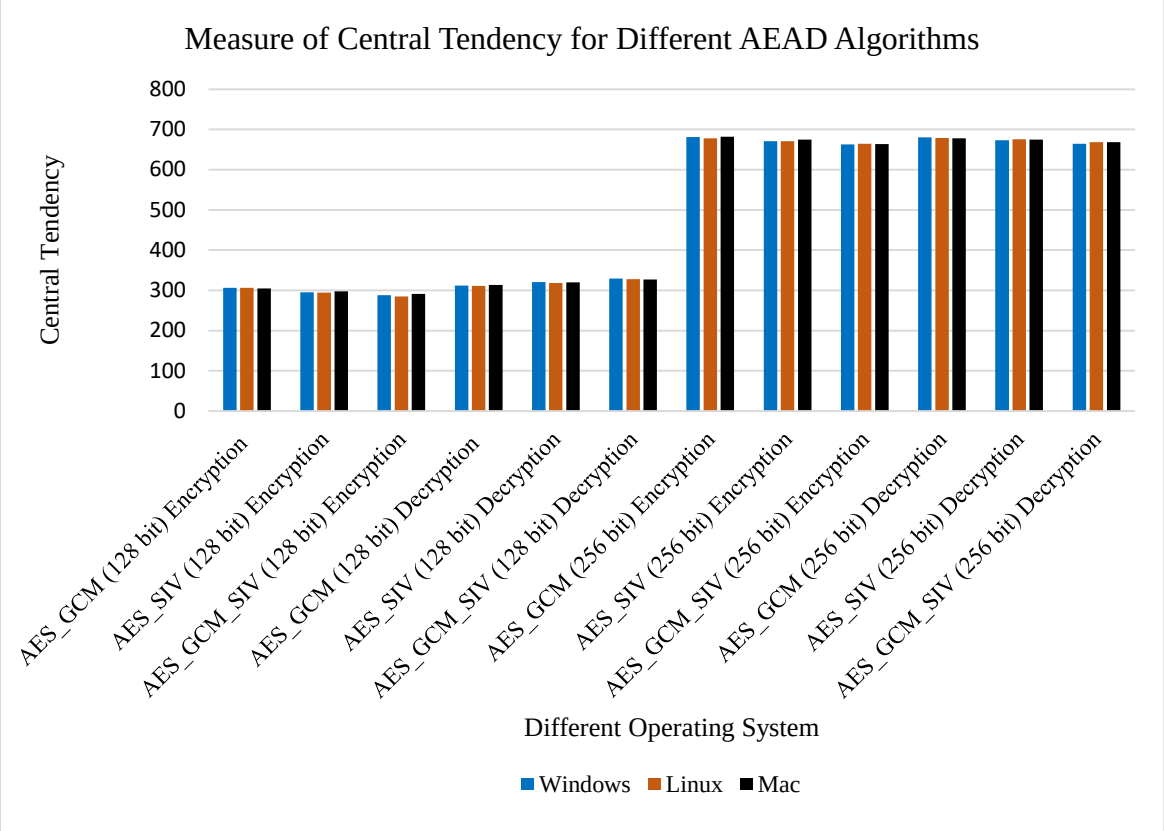

Figure 3: Comparison of Measure of central tendency of different AEAD algorithms with three operating system

5.2. Two Way ANOVA

The two way ANOVA (Analysis of Variance), also known as two-factor ANOVA used to determine if two or more samples have the same "mean" or average. Anova is a technique of understanding the variance of variables. The two way ANOVA is measured on the basis of AEAD algorithm encryption/decryption with two key size 128-bit as type 1 and 256-bit as type 2 from Table 1. It makes it possible to calculate how much a particular variable affects the final result. Anova technique does this by eliminating or confirming the null hypothesis. A null hypothesis means that there exists no relationship at all between the two entities under observation. The significance of a particular variable or entity is calculated by comparing the values with the overall impact on the target value. Anova requires a certain number through which it can analyze the null hypothesis that we pose at the start of the analysis. The three critical values for this calculation are F ratios and F-critical, with some significance values. For example, X's significance will be more on A, if even a small change in X can affect in changing the value of A. The F ratios are calculated by the Mean sum of squares of an entity and the mean sum of residuals squares. The mean sum of squares is calculated by dividing the mean sum of squares by the degree of freedom. The degree of freedom is the number of possible cases of the nominal variable, minus one. F critical is based on the significance values. F ratios are calculated manually through the process explained above. The validity of the hypothesis is dependent on the values of F ratios and F critical. Here are the two cases:

- If the F-critical > F ratio, then the hypothesis holds, and there is no relation between the variables under observation

- If the F-critical < F ratio, then the hypothesis can be declared invalid, and in turn, supports the idea that the variables affect each other.

The purpose of ANOVA is to measure the differences in strength of AEAD algorithms with different key sizes.

Null Hypothesis: The null hypothesis states that there is no relationship between two population parameters, i.e., independent and dependent variables. If the hypothesis shows a relationship between the two parameters, the outcome could be due to an experimental or sampling error. However, if the null hypothesis returns false, there is a relationship in the measured phenomenon. The null hypothesis is helpful because it can be tested to conclude whether or not there is a relationship between two measured phenomena. It can inform the user whether the results obtained are due to chance or manipulating a phenomenon. Testing a hypothesis sets the stage for rejecting or accepting a hypothesis within a certain confidence level.

The overall average for the two different key sizes (128-bit and 256-bit) in the data is shown in Table 4. The difference is about 363.5333. The average for 5 different message sizes has been shown in a total average of two factors with replication. The averages for message sizes 128-bit, 1024-bit, 2048-bit, 4096-bit and 8192-bit are 318, 408.1667, 495.1667, 570.3333, and 660.1667.

Suppose the key size from the different message sizes all works the same on both key sizes (type 1 and type 2). In that case, the averages for the individual groups should follow the same patterns: The average for key size (type 1) should be higher, and the average for message size 8192-bit should be higher.

The group averages show a different pattern than the overall averages for the two factors. Message size 8192-bit‘s average is higher than the other four because there is a more significant difference between the message sizes for the second key size (type 2).

The comparison of averages should prepare us for what to expect about the null hypothesis for two-way ANOVA that the factors do not affect the response variable.

Table 4: The measure of Analysis of variance using ANOVA: two-factor with replication

Anova: Two-Factor With Replication						
SUMMARY	**128-bit**	**1024-bit**	**2048-bit**	**4096-bit**	**8192-bit**	**Total**
type 1 (128-bit)						
Count	6	6	6	6	6	30
Sum	1510	1599	1930	2047	2172	9258
Average	251.6667	266.5	321.6667	341.1667	362	308.6
Variance	17.46667	24.3	827.4667	776.5667	361.6	2228.179
type 2 (256-bit)						
Count	6	6	6	6	6	30
Sum	2306	3299	4012	4797	5750	20164
Average	384.3333	549.8333	668.6667	799.5	958.3333	672.1333
Variance	31.86667	62.56667	357.0667	43.1	49.46667	40631.22

Total						
Count	12	12	12	12	12	
Sum	3816	4898	5942	6844	7922	
Average	318	408.1667	495.1667	570.3333	660.1667	
Variance	4822.545	21933.42	33377.24	57664.24	97172.33	
ANOVA						
Source of Variation	***SS***	***df***	***MS***	***F***	***P-value***	***F crit***
Sample	1982347	1	1982347	7769.442	1.57E-56	4.03431
Columns	860602.3	4	215150.6	843.2427	3.33E-45	2.557179
Interaction	369563.1	4	92390.77	362.1085	3.05E-36	2.557179
Within	12757.33	50	255.1467			
Total	3225270	59				

For the two-way ANOVA, our largest p-value is about $1.57 * 10^{-56}$. That is much smaller than the traditional cutoff value for statistical significance of 0.05.

Because the p-value for the interaction is small, we cannot make a simple statement that one key size leads to a higher strength in terms of robustness.

The hypothesis test confirms, what we might have expected from the examination of averages: The effect of the different AEAD algorithms depends on the key size (128-bit and 256-bit).

6. Conclusion

Authenticated encryption algorithms are building blocks of secure communication over the network. In this paper, statistical analysis of three different algorithms with the key size 128/256 bit for a message of block size 128/1024/2048/4096/8192 bit with three different operating systems in a virtual environment has been observed. The overall performance of AES_GCM_SIV 128/256 bit encryption is faster than AES_GCM, AES_SIV and has a slight percentage error concerning central tendency. Here AES_GCM with key size 128 bit is faster than AES_SIV and AES_GCM_SIV and has a minor percentage error. However, decryption of AES_GCM_SIV 256 bit is faster than AES_GCM and AES_SIV and have minor percentage error on almost every operating system. So the AES_GCM_SIV 128/256 bit is the best encryption algorithm than AES_GCM and AES_SIV, while decryption of AES_GCM 128 bit and AES_GCM_SIV 256 bit are the better algorithms. From the result ANOVA two factor with replication, we conclude that the P-value for the interaction is small. So, we cannot make a simple statement that one key size leads to a higher strength in terms of robustness. The statistical analysis of other AEAD algorithms also can be compared and fine-tuned for better results in the future.

References

Hell, M., Johansson, T. and Brynielsson, L. (2009). An overview of distinguishing attacks on stream ciphers. *Cryptography and Communications*, **1(1)**, 71-94.

Junod, P. and Vaudenay, S. (2003). Optimal key ranking procedures in a statistical cryptanalysis. *In International Workshop on Fast Software Encryption, Springer, Berlin, Heidelberg*, 235-246.

Junod, P. (2003). On the optimality of linear, differential, and sequential distinguishers. *In International Conference on the Theory and Applications of Cryptographic Techniques*, *Springer, Berlin, Heidelberg*, 17-32.

Kannojia, S. P. and Kurmi, J. (2021). Attack Detection Framework for Authenticated Encryption cipher: An Experimental Approach. *Shodh Sarita*, **8(29)**, 210-215.

Kannojia, S. P. and Kurmi, J. (2021). Comparative Study of SSL/TLS Cryptographic Libraries. *International Journal of Innovative Research in Science, Engineering and Technology*, **10(8)**, 11658-11662.

Kannojia, S. P. and Kurmi, J. (2021). Analysis of Cryptographic Libraries(SSL/TLS). *International Journal of Computer Sciences and Engineering*, **9(9)**, 59-62.

Knuth, D. E. (2014). Art of computer programming. *Semi-Numerical Algorithms. Addison-Wesley Professional*, Volume **2**.

Maurer, U. M. (1992). A universal statistical test for random bit generators. *Journal of Cryptology*, **5(2)**, 89-105.

Prouff, E., Rivain, M. and Bevan, R. (2009). Statistical analysis of second order differential power analysis. *IEEE Transactions on computers*, **58(6)**, 799-811.

Vaudenay, S. (1996, January). An experiment on DES statistical cryptanalysis. *In Proceedings of the 3rd ACM Conference on Computer and Communications Security* (139-147).