ISBN 978-981-11-3671-9
Proceedings of 2017 the 7th International Workshop on Computer Science and Engineering (WCSE 2017)
Beijing, 25-27 June, 2017, pp. 496-502
doi: 10.18178/wcse.2017.06.086

A Cost analysis of AES-128 and AES-512 on Apple mobile processors

Vatchara Saicheur [+], and Krerk Piromsopa

Department of Computer Engineering, Chulalongkorn University, Bangkok, Thailand

Abstract. This paper is the cost analysis of two Advanced Encryption Standard (AES) algorithms on 32-bit and 64-bit Apple mobile processor by using iPhone5C and iPhone7. Our analysis shows increasing in performance when expanding the block size from 128 bits (AES-128) to 512 bits (AES-512). Similarly increasing the length of encryption key to 512 bits and 1024 bits yields stronger security. Our aim is to analyze the encryption cost different between the original AES-128 and the AES-512. The results showed that increasing block size will give 1.21 – 1.64 speed up on iPhone5C and 1.19 – 1.55 speed up on iPhone7 depending on the key length. Moreover, AES 512-bit block size shows faster CPU time in encryption than 128-bit block size. However, the memory usage for encryption on all key size are similar. iPhone7 used more memory than iPhone5C. In conclusion, expanding block size to 512 bits can increase performance while this is also lower the cost on mobile device. This result may have the benefit in improving the security of personal data by using mobile phone

Keywords: Cryptography, Cost analysis, Mobile devices

1. Introduction

Today, the Internet and technology has made our life more comfortable. Many devices are designed to help facilitate comfortable in everyday life such as smartphone, tablet, smart watch and computer. Those devices have high specification and can be used for many things. When people are using mobile devices for communication or data transmission through the internet network, it is also increase the risk of data leaked or data stolen by third party. Particularly, if important data was stolen, it may cause serious damage. A tool for protecting our data is encryption.

There are many ways to encrypt the data. The popular standard encryption is AES algorithm. The AES algorithm is a standard encryption algorithm that has been certified by the US government for encrypting and storing personal information. It is a symmetric key algorithm which has been accepted and presented by NIST since 2001[1]. However, implementing AES algorithm on mobile devices can be difficult and resource intensive. If we can improve the performance and can lower the resource utilization at the same time, it will allow applications to encrypt data on mobile device instead of off-loading to another computer.

This paper proposed to expand the AES block size to 512 bits for using in mobile devices. To promote this concept, we compare the cost between AES 128 bits and the new block size by varying different key sizes: 128, 192, 256, 512 and 1024 bits. To make our analysis more complete, the experiment is based on both 32-bit and 64-bit mobile architecture (Apple ARM A6 and Apple ARM A10 respectively).

2. Related Work

Cryptography is the study of techniques for securing information in the presence of third parties such as microdots, steganography (embedding bits in an image), and other ways to hide data. Effective cryptography is associated with the development and the creation of mathematical algorithms[2].

[+] Corresponding author. Tel.: +66866043223.
E-mail address: turnofmut@gmail.com.

2.1. AES-128 algorithm

AES is a symmetric-key encryption using the array of 4x4 of 128-bit ciphering block. It uses the key size of 128, 192, and 256 bits with 10, 12, or 14 iteration rounds, respectively. There are four major operations; byte substitution, row shifting, column mixing and adding round key [3], [4]. It is widely adopted and supported by many applications. It considered secure compared to early algorithms such as Data Encryption Standard (DES) which is certified by Federal Information Processing Standard (FIPS)[5].

2.2. AES-512 implementation

Several AES implementations were present in many studies. The difference applications of AES require different implementations of same algorithm. The FPGA architecture for a new version of the Advanced Encryption Standard (AES) algorithm[6] proposed the implementation of AES using input block size and key size of 512 bits developed in VHDL, and synthesized using Virtix-6 and Virtex-7 chips. The study found that this design obtained high level of security to cryptanalysis and yields the speed up of 2.3 comparing to the original AES-128 implementation[7].

3. Proposed Methodology

This study contains 2 steps. First is the implementation of AES for using on mobile devices with the expansion of block size to AES-512. Second is the comparison of efficacy and the cost of using algorithm by comparing speed up, CPU time and memory utilization on 32-bis and 64-bit Apple mobile processors. The specification of the processor can be described in Table I.

Table I: Specification of iPhone5C and iPhone7

Model	**iPhone5C**	**iPhone7**
CPU	ARM A6 32-bit architecture Dual-core processor 1.3 GHz	ARM A10 64-bit architecture Quad-core processor 2.34 GHz
Ram	1 GB	2 GB
iOS	10.2	10.2

The implementation of AES-512 algorithm can be explained as follow:

3.1. Block size expansion

The block state of the original AES-128 is an 4x4 array. Each row and column of input with Nb byte requires Nb = 4. In this paper, we increase the block state to 8x8 array. This also increase the input data in required Nb = 8 as shown in Fig. 1.

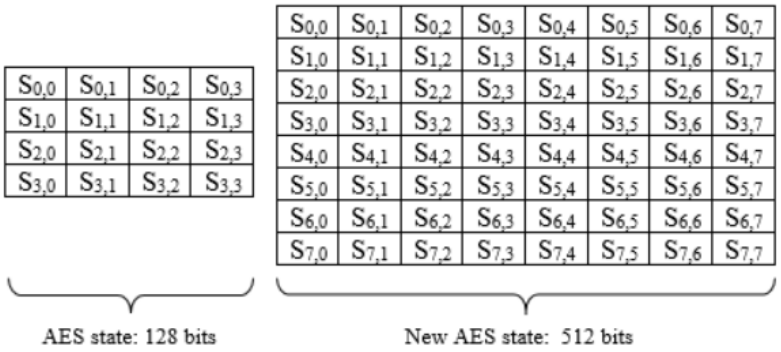

Fig. 1: The block state of original AES-128 and new AES-512 bits.

3.2. Byte Substitution

The plaintext Input are assigned in the array 8x8 of 64 bytes. Each byte is independently substituted by the values by the substitution boxes (similar to that of the AES-128) to increase the complexity and the security of algorithm as shown in Fig. 2.

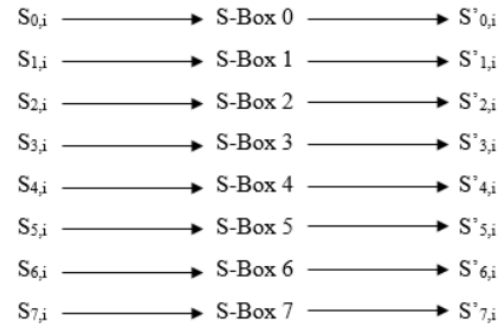

Fig. 2: The byte substitution of AES-512 bits.

3.3. Row shifting

After the byte substitution process, the resulting matrix rows are shifted with simple process. The bytes in each row in the matrix will be rotated to the left. The number of left rotations will be increased by one per row. In this case, the zeroth row is not shifted; the first row is shifted by one byte; the second row is shifted by two bytes. The increasing is repeated until the last row as shown in Fig. 3.

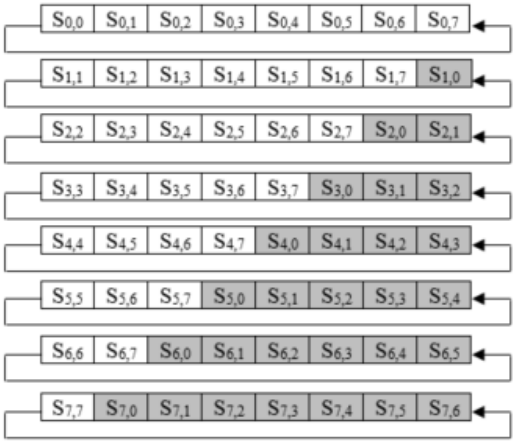

Fig. 3: The row shifting of AES-512 bits.

3.4. Column mixing

After the shifting row process, the result is the bit changed in the state block. Next process is converting data in columns by multiplying with a pre-defined polynomial. Each column contains 8 bytes of data which must be applied to a column of data matrix. Column mixing are based on the concept of polynomial over GF(2^n). The columns in the data matrix (shown in Fig. 4) will be multiplied by a fixed polynomial of a(x) as given in (1).

$$a(x) = [02]x^7 + [01]x^6 + [03]x^5 + [01]x^4 + [01]x^3 + [01]x^2 + [01]x^1 + [01]x^0 \quad (1)$$

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \\ S'_{4,c} \\ S'_{5,c} \\ S'_{6,c} \\ S'_{7,c} \end{bmatrix} = \begin{bmatrix} 02 & 01 & 03 & 01 & 01 & 01 & 01 & 01 \\ 01 & 03 & 01 & 01 & 01 & 01 & 01 & 02 \\ 03 & 01 & 01 & 01 & 01 & 01 & 02 & 01 \\ 01 & 01 & 01 & 01 & 01 & 02 & 01 & 03 \\ 01 & 01 & 01 & 01 & 02 & 01 & 03 & 01 \\ 01 & 01 & 01 & 02 & 01 & 03 & 01 & 01 \\ 01 & 01 & 02 & 01 & 03 & 01 & 01 & 01 \\ 01 & 02 & 01 & 03 & 01 & 01 & 01 & 01 \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \\ S_{4,c} \\ S_{5,c} \\ S_{6,c} \\ S_{7,c} \end{bmatrix}$$

Fig. 4: The data matrix for column mixing encryption of AES-512 bits.

The multiplication result is modulo by p(x)= x8 + 1 to maintain the resulting polynomial with the degree of less than 8. The inversing of column mixing as shown in Fig. 5 will be multiplied with the inverse of the polynomial a'(x) as given in (2).

$$a'(x) = [0E]x^7 + [01]x^6 + [09]x^5 + [01]x^4 + [0D]x^3 + [01]x^2 + [0B]x^1 + [01]x^0 \quad (2)$$

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \\ S'_{4,c} \\ S'_{5,c} \\ S'_{6,c} \\ S'_{7,c} \end{bmatrix} = \begin{bmatrix} 0e & 01 & 09 & 01 & 0d & 01 & 0b & 01 \\ 01 & 09 & 01 & 0d & 01 & 0b & 01 & 0e \\ 09 & 01 & 0d & 01 & 0b & 01 & 0e & 01 \\ 01 & 0d & 01 & 0b & 01 & 0e & 01 & 09 \\ 0d & 01 & 0b & 01 & 0e & 01 & 09 & 01 \\ 01 & 0b & 01 & 0e & 01 & 09 & 01 & 0d \\ 0b & 01 & 0e & 01 & 09 & 01 & 0d & 01 \\ 01 & 0e & 01 & 09 & 01 & 0d & 01 & 0b \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \\ S_{4,c} \\ S_{5,c} \\ S_{6,c} \\ S_{7,c} \end{bmatrix}$$

Fig. 5: The data matrix for column mixing decryption of AES-512 bits.

3.5. Add round key

In this process, each byte of key will be added to data state in each round of encryption process. The setting key is required to expand before used. The expanded key will be XORed with each state column until the last column of each round as shown in (3). A round number of encryption is different for each key size as shown in Table II.

$$[S'_{0,c}, S'_{1,c}, S'_{2,c}, \ldots, S'_{7,c}] = [S_{0,c}, S_{1,c}, S_{2,c}, \ldots, S_{7,c}] \oplus [w_{round*Nb+c}] \quad (3)$$

Table II: Round Number of AES-128 bits and AES-512 bits

Key size	Round Number AES-128 bits	Round Number AES-512 bits
128 bits	10	4
192 bits	12	5
256 bits	14	6
512 bits	22	10
1024 bits	38	18

3.6. Key expansion

The key expansion is the process of preparing key for each round of encryption and decryption process. The key will be kept in words of eight bytes each. Words is defined by the number of column used for each key size. It is often written in the form of number of key (Nk). The algorithm will generate sub-keys Nb(Nr+1) words, they depend on the number of block (Nb) and the number of round (Nr). This process requires SubWord() and RotWord() functions. SubWord() receives input words of 8 bytes for substituted by the values in the substitution boxes. RotWord() receives input word [a0, a1, a2, a3, a4, a5, a6, a7], to be rotated left to [a1, a2, a3, a4, a5, a6, a7, a0]. Moreover, the round constant (Rcon[i]) should be used as defined by (4):

Let: i be the round number

$$Rcon[i] = 00000010^{(i-8)/8} \quad (4)$$

The round key expansion is performed according to the following equations:

If i is a multiple of Nk

$$w[i] = (SubWord(RotWord(w[i-1])) \oplus Rcon[i/Nk]) \oplus w[i-Nk] \quad (5)$$

If i is not a multiple of Nk

$$w[i] = w[i-1] \oplus w[i-Nk] \quad (6)$$

3.7. Decryption process

The encrypted cipher should be inverted by starting with InvertShiftRows(), InvertSubBytes(), AddRoundKey() and InvertMixColumns() respectively.

4. Experimental results and evaluation

In this study, we have implemented AES-128 and AES-512 on mobile phone by using iPhone5C and iPhone7. The results from 2-type of AES show that: AES-512 algorithms have better speed than AES-128 algorithms. The speed up depends on devices. Our experimental results showed that iPhone5C has the maximum speed up at the encrypted data size of 256 kb. However, iPhone7 has the maximum speed up at encrypted data size 1024 KB as shown in Fig. 6.

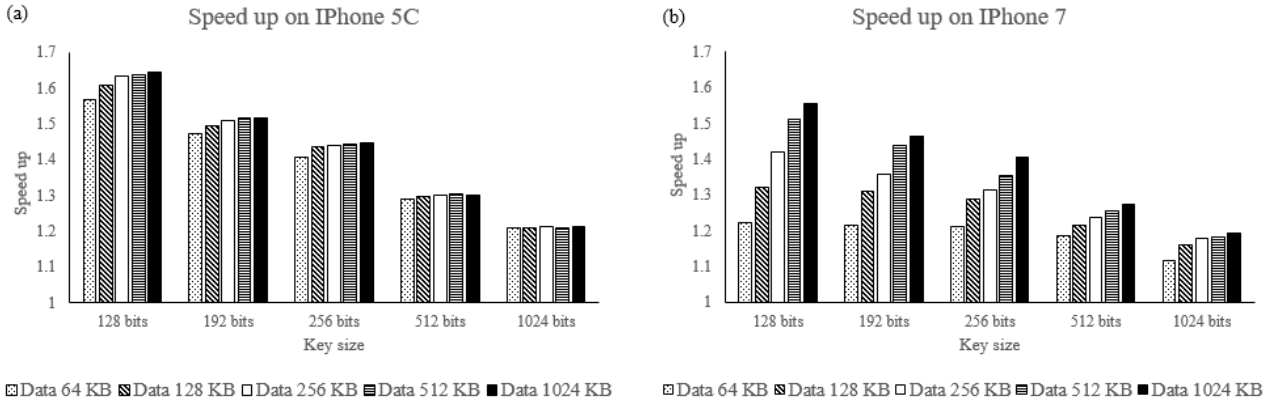

Fig 6: Speed up testing of key size 128, 192, 256, 512 and 1024 bits compare between AES-512/AES-128 algorithms on iPhone5C (a) and iPhone7 (b) at encrypted data size 64, 128, 256, 512 and 1024 KB.

The AES-512 algorithms on iPhone5C yield about 1.21 – 1.64 speed up comparing to those of AES-128 algorithm. The AES-512 algorithms on iPhone7, like iPhone5c, yield about 1.19 – 1.55 speed up as shown in Fig. 7.

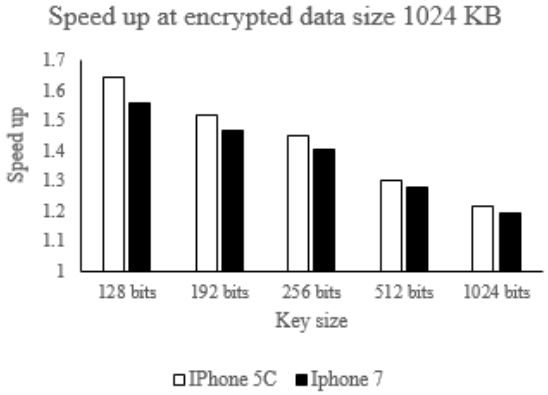

Fig. 7: Speed up testing of key size 128, 192, 256, 512 and 1024 bits compare between AES-512/AES-128 algorithms on iPhone5C and iPhone7 at encrypted data size 1024 KB.

Table III: Speed up between AES-512 bits/AES-128 bits

Key size	**Speed up**	
	iPhone5C	**iPhone7**
128 bits	1.64	1.55
192 bits	1.51	1.46
256 bits	1.45	1.40
512 bits	1.30	1.27
1024 bits	1.21	1.19

The cost analysis of CPU usage found that CPU will run at full speed when algorithms are being processed. The AES-512 algorithms use less CPU times than that of the AES-128 about 2,008 – 3,584 milliseconds on iPhone5C. Increasing the key size will result in higher different in CPU time. iPhone7 showed smaller different in CPU time. The AES-512 use less CPU time than that of the AES-128 about 330 – 550 milliseconds. The result is shown in Fig. 8.

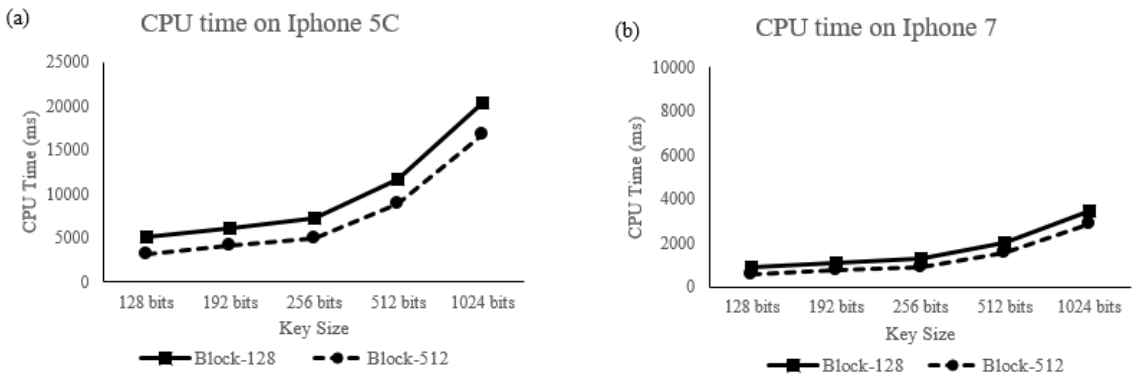

Fig. 8: CPU time testing of key size 128, 192, 256, 512 and 1024 bits compare between AES-512/AES-128 algorithms on iPhone5C (a) and iPhone7 (b) at encrypted data size 1024 KB.

Table IV: CPU times of AES-128 bits and AES-512 bits

Devices	Key size (bits)	CPU Time (ms)		
		AES-128	AES-512	Different time
iPhone5C	**Key 128**	5134	3126	2008
	Key 192	6207	4099	2108
	Key 256	7328	5060	2268
	Key 512	11664	8959	2705
	Key 1024	20356	16772	3584
iPhone7	**Key 128**	914	588	326
	Key 192	1106	756	350
	Key 256	1287	916	371
	Key 512	1993	1563	430
	Key 1024	3440	2887	553

The analysis of memory usage shows that at the same size of encrypted data, memory usage of both algorithms is similar. However, the iPhone7 uses memory more than that of iPhone5C as shown in Fig. 9.

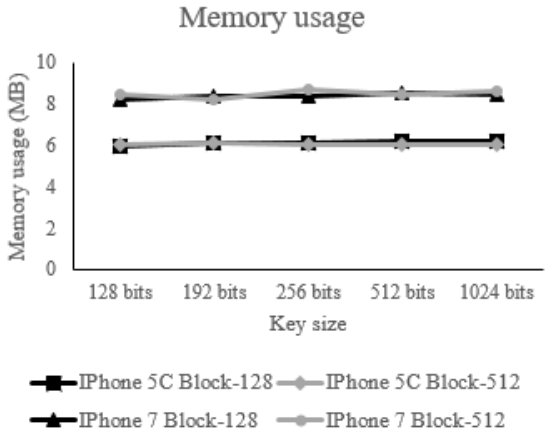

Fig. 9: Memory usage testing of key size 128, 192, 256, 512 and 1024 bits compare between AES-512/AES-128 algorithms on iPhone5C and iPhone7 at encrypted data size 1024 KB.

Table V: Memory usage of AES-128 bits and AES-512 bits

Devices	Key size (bits)	Memory usage (MB)	
		AES-128	AES-512
iPhone5C	**Key 128**	5.97	6.07
	Key 192	6.13	6.10
	Key 256	6.13	6.06
	Key 512	6.20	6.00
	Key 1024	6.20	6.06
iPhone7	**Key 128**	8.17	8.46
	Key 192	8.40	8.17

	Key 256	8.33	8.67
	Key 512	8.53	8.46
	Key 1024	8.46	8.60

5. Conclusion

In conclusion, we have proposed the use of AES-512 algorithm supported on mobile devices. Our study showed the cost analysis between the original AES-128 and the new AES-512 algorithms with various key size. The new algorithm showed higher speed than the original AES at lower cost. Although, encryption with longer length of key showed less difference than that of the shorter key, the new algorithm provides stronger security. Thus it is more useful for data protection.

When comparing between 32-bit and 64-bit CPU, (iPhone5C and iPhone7), we found that both devices give the same trend for both algorithms.

With stronger level of security and better performance, we expect that the use of AES-512 algorithm would be useful for people and organizations in protecting their important data.

6. References

[1] Federal Information Processing Standards (FIPS), "Announcing the Advanced Encryption Standard (AES)," *National Institute of Standards and Technology (NIST)*, 2001.

[2] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1996.

[3] J. Daemen and V. Rijmen, *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer Science & Business Media, 2002.

[4] T. Jamil, "The Rijndael algorithm," *IEEE Potentials*, vol. 23, no. 2, pp. 36–38, Apr. 2004.

[5] Federal Information Processing Standards (FIPS), "Data Encryption Standard (DES)," *National Institute of Standards and Technology (NIST)*, 1999.

[6] V. Arun, K. Vanisree, and D. L. Reddy, "Implementation of AES-GCM encryption algorithm for high performance and low power architecture Using FPGA," *Proc. ICETET*, vol. 29, p. 30th, 2014.

[7] A. Moh'd, Y. Jararweh, and L. 'ai Tawalbeh, "AES-512: 512-bit Advanced Encryption Standard algorithm design and evaluation," in *Information Assurance and Security (IAS), 2011 7th International Conference on*, 2011, pp. 292–297.