

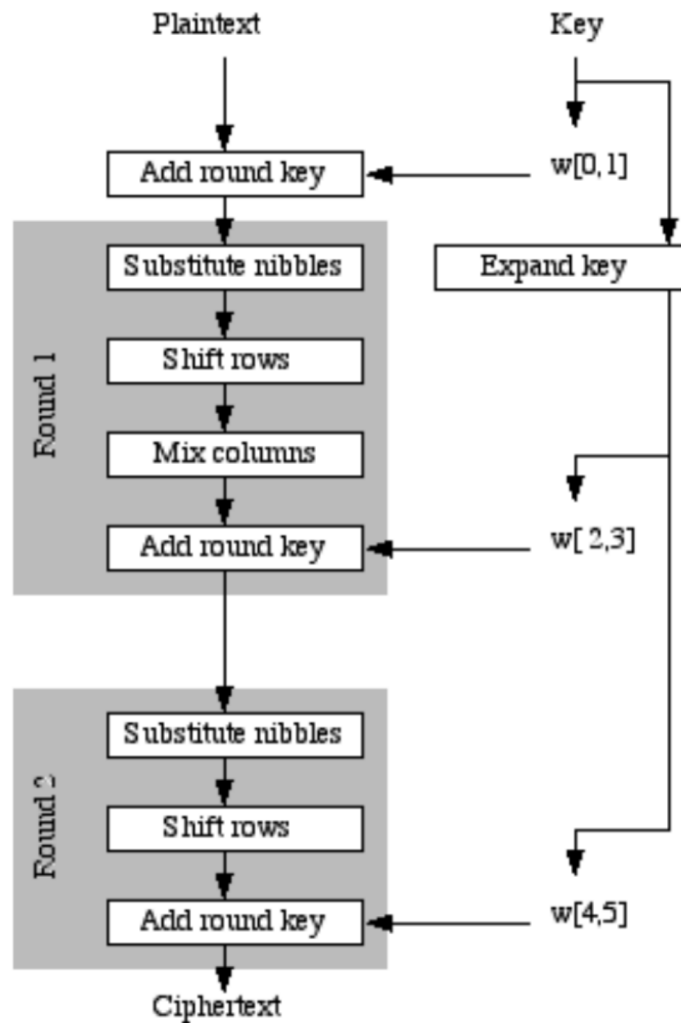
A Simplified AES Algorithm

Presented by Joshua Holden, Rose-Hulman Institute of Technology

Figures by Lana Holden

Algorithm invented by Mohammad Musa, Edward Schaefer, and Stephen Wedig

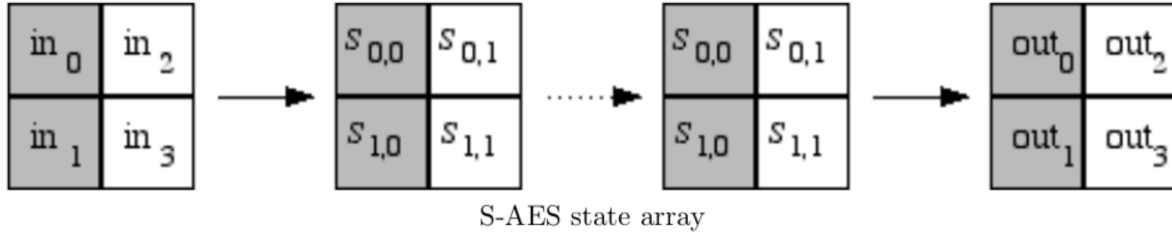
Overview S-AES is to AES as S-DES is to DES. In fact, the structure of S-AES is exactly the same as AES. The differences are in the key size (16 bits), the block size (16 bits) and the number of rounds (2 rounds). Here is an overview:



S-AES Encryption Overview

Substitute nibbles Instead of dividing the block into a four by four array of bytes, S-AES divides it into a two by two array of “nibbles”, which are four bits long. This is called the state

array and is shown below.



In the first stage of encryption, an S-box is used to translate each nibble into a new nibble. First we associate the nibble $b_0b_1b_2b_3$ with the polynomial $b_0x^3 + b_1x^2 + b_2x + b_3$. This polynomial is then inverted as an element of $GF(16)$, with the “prime polynomial” used being $x^4 + x + 1$. Then we multiply by a matrix and add a vector as in AES.

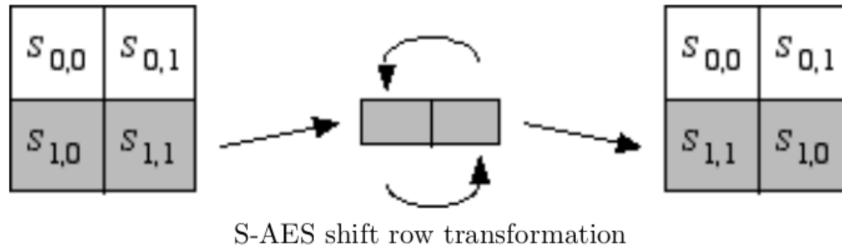
$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Remember that the addition and multiplication in the equation above are being done modulo 2 (with XOR), but not in $GF(16)$.

Since a computer would do the S-box substitution using a table lookup, we give the full table for the S-box here.

nibble	S-box(nibble)	nibble	S-box(nibble)
0000	1001	1000	0110
0001	0100	1001	0010
0010	1010	1010	0000
0011	1011	1011	0011
0100	1101	1100	1100
0101	0001	1101	1110
0110	1000	1110	1111
0111	0101	1111	0111

Shift Rows The next stage is to shift the rows. In fact, the first row is left alone and the second row is shifted.



Mix Columns After shifting the rows, we mix the columns. Each column is multiplied by the matrix

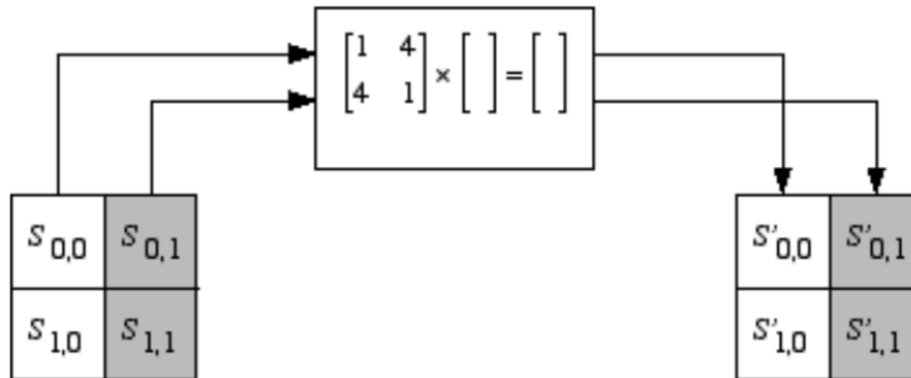
$$\begin{bmatrix} 1 & 4 \\ 4 & 1 \end{bmatrix}.$$

These operations **are** done in GF(16), so remember that the 1 corresponds to the polynomial 1 and the 4 corresponds to the polynomial x^2 . Thus this matrix could also be written as

$$\begin{bmatrix} 1 & x^2 \\ x^2 & 1 \end{bmatrix}.$$

Don't forget to reduce modulo $x^4 + x + 1$.

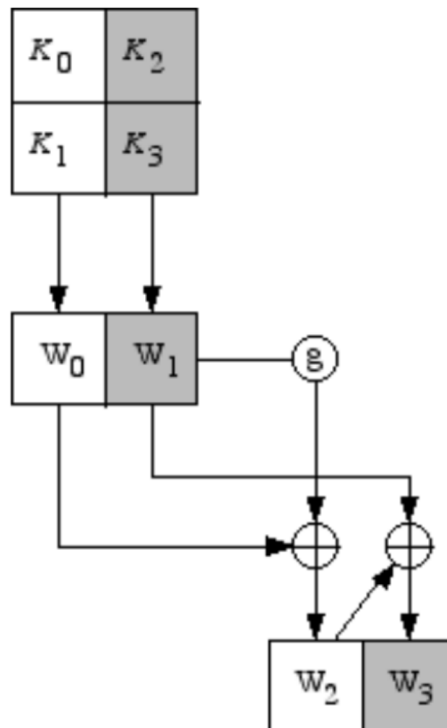
The mix column transformation is omitted in the last round, in order to simplify the decryption.



S-AES mix column transformation

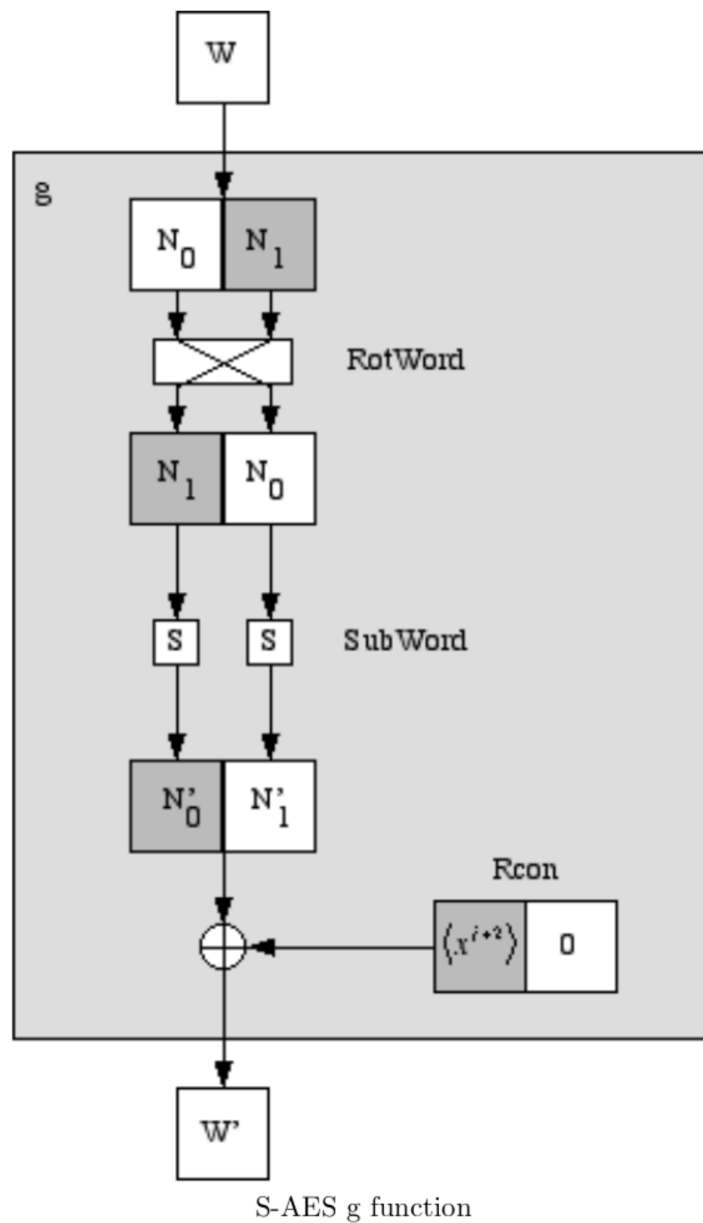
Add Round Key The last stage of each round of encryption is to add the round key. (In fact, this is also done before the first round.) Before the first round, the first two words (W_0 and W_1) of the expanded key are added. In the first round, W_2 and W_3 are added. In the last round, W_4 and W_5 are added. All additions are done modulo 2, that is, with XOR.

Key Expansion Key expansion is done very similarly to AES. The four nibbles in the key are grouped into two 8-bit “words”, which will be expanded into 6 words. The first part of the expansion, which produces the third and fourth words, is shown below. The rest of the expansion is done in exactly the same way, replacing W_0 and W_1 with W_2 and W_3 , and replacing W_2 and W_3 with W_4 and W_5 .



S-AES key expansion

The g function is shown in the next diagram. It is very similar to AES, first rotating the nibbles and then putting them through the S-boxes. The main difference is that the round constant is produced using x^{j+2} , where j is the number of the round of expansion. That is, the first time you expand the key you use a round constant of $x^3 = 1000$ for the first nibble and 0000 for the second nibble. The second time you use $x^4 = 0011$ for the first nibble and 0000 for the second nibble.



Exercise Use the key 1010 0111 0011 1011 to encrypt the plaintext “ok” as expressed in ASCII, that is 0110 1111 0110 1011. The designers of S-AES got the ciphertext 0000 0111 0011 1000. Do you?

References

- [1] Mohammad Musa, Edward Schaefer, and Stephen Wedig, *A simplified AES algorithm and its linear and differential cryptanalyses*, Cryptologia **27** (April 2003), 148–177.