Quasigroups and Related Systems **13** (2005), 265 – 268

AES with the increased confidentiality

Czesław Kościelny

Abstract

It has been shown in the paper how to use well known encrypting algorithms AES -128, AES-192 and AES-256 as algorithms AES-340, AES-404 and AES-468, respectively, having considerably increased key space.

1. Introduction

As it is known, the AES algorithm [1] is a symmetric-key block cipher which uses cryptographic keys of 128, 192 and 256 bits to encrypt and decrypt data in blocks of 128 bits. From the mathematical point of view this algorithm is interesting for any algebraist, as an example of advanced computing in Galois fields. In particular, the algorithm apply one fixed element from $GF(256)$ to compute round constant array, one affine transformation over $GF(2)$ with fixed 8×8 matrix and c vector, a fixed pair of mutually invertible polynomials of degree $\leqslant 3$ over $GF(256)$ belonging to the polynomial ring modulo x^4-1, a fixed irreducible polynomial of degree 8 over $GF(2)$ defining multiplication in $GF(256)$, and performs many operations of multiplication, addition and inversion in this field. Furthermore, $GF(256)$ can be perceived not only as a field but simultaneously as two groups, two quasigroups or two groupoids as well. That is why this paper deserves, in the author's opinion, to be published in Quasigroups and Related Systems, even though it concerns highly application oriented problem.

2000 Mathematics Subject Classification: 68P25, 11T
Keywords: cryptography, AES algorithm, Galois fields.

2. A method of using the AES algorithm with considerably enlarged key space

Without going into details we may shortly say, that in order to use the algorithm AES [1] as a cryptosystem with the increased confidentiality it simply suffices to replace all fixed constants, appearing in cryptographic transformations and routines of the algorithm, viz.

▷ the value `{02}` in `Rcon[i][1]`,

▷ the elements of affine transformation, i.e. the byte value $c =$ `{63}` and the matrix

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix},$$

▷ the non-primitive irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$ over $GF(2)$,

▷ the polynomial $a(x) =$ `{03}`$x^3 +$ `{01}`$x^2 +$ `{01}`$x +$ `{02}` over $GF(256)$ and its inverse modulo $x^4 - 1$,

▷ and finally the number of rounds r that should be taken into account during the execution of `KeyExpansion`, `Cipher` and `InvCipher` routines,

▶ by 6 variables, stored in the array

$$K_S = [K_{\texttt{Rcon}}, K_c, K_A, K_m, K_a, K_r], \tag{1}$$

which will form together with 128, 192 or 256 bit key K a cryptographic key for the generalized in this way AES:

$$K_{IC} = K,\, K_S. \tag{2}$$

It should be explained in this place that the sufficient level of diffusion and confusion may be attained already after executing from one to three rounds of encrypting procedure, therefore, the value of actual parameter replacing the formal parameter Nr in the invocation statement of the routines `Cipher` and `InvCipher`, viz. the element K_r of the array K_S may belong to the interval $[3, Nr]$.

The presented approach allows the reader either to design a large class of AES ciphers with various cryptographic transformations and rule of multiplication in $GF(256)$, or to use the AES algorithms as a quite strong cipher with 7-element cryptographic key. Considering the second case of this alternative, one should first determine an equivalent increment of the cryptographic key length of the cryptosystem resulting from the presented approach. This increment can be computed by means of the equation

$$\Delta_K = \lfloor \frac{\ln(256^2 \cdot (Nr-2) \cdot Nip \cdot Nnsm_{8\times 8}(2) \cdot Nnscm_{4\times 4}(256))}{\ln(2)} \rfloor, \quad (3)$$

where the formulae

$$Nnsm_{8\times 8}(q) = \prod_{k=0}^{n-1} (q^n - q^k), \quad Nnscm_{4\times 4}(q) = \frac{(q-1)^4\, q^{12}}{4},$$

determine the number of non-singular matrices 8×8 over $GF(q)$ (see [3, p. 3]) and the number of non-singular circulant matrices 4×4 over $GF(q)$ of characteristic 2 (see [3, pp. 7, 80]) (equal also to the number of invertible modulo x^4-1 polynomials of degree $\leqslant 3$ over $GF(256)$), respectively,
Nip – denotes the number of irreducible polynomials of degree 8 over $GF(2)$,
Nr – equals to the number of rounds depending on the key length as is recommended by [1].

Taking into account [1], (3) and the fact that the $Nip = 30$, we get $\Delta_K = 212$, which means that we can use AES-128, AES-192 and AES-256 as AES-128+Δ_K, AES-192+Δ_K and AES-256+Δ_K, that is as AES-340, AES-404 and AES-468, correspondingly. To implement AES-340, AES-404 and AES-468 we may bring into play almost the same software or hardware as for AES-128, AES-192 and AES-256. Assuming that the implementation of described here AES algorithm with the increased level of privacy will explicitly employ the operations in $GF(256)$, namely, addition, multiplication and rising to a positive or negative power, performed by means of the appropriate routines, arrays or hardware, we may summarize the encrypting/decrypting procedures as follows:
Encryption: Entity $\mathcal{B}$ encrypts a plaintext block M for entity $\mathcal{A}$, which $\mathcal{A}$ decrypts. Thus $\mathcal{B}$ should make the following steps:
STEP 1: Generate the cryptographic key $K_{IC} = K$, K_S (K_{IC} may be used for encryption the whole message, consisting of many blocks). The elements of the array K_S should be, of course, suitably computed.
STEP 2: Send the key K_{IC} to $\mathcal{A}$ using secure channel.

STEP 3: Adapt the system for computing in $GF(256)$ defined by the polynomial K_m contained in K_S, compute S-box and inverse S-box tables and the the array `Rcon` taking into account K_A, K_c and $K_{\texttt{Rcon}}$, next modify all these algorithm's transformations and routines given in [1], which depend on data stored in K_S.
STEP 4: Generate the key schedule using the the same `KeyExpansion` routine as in [1], but apply the value K_r as the actual parameter of the formal parameter Nr.
STEP 5: Compute the ciphertext block C of the plaintext block M using the same routine `Cipher` as given in [1], but apply the value K_r as the actual parameter of the the formal parameter Nr, next send the ciphertext to $\mathcal{A}$ thorough unsecured channel.
Decryption: To find plaintext block M from the ciphertext block C, the entity $\mathcal{A}$ should perform the following operations:
STEP 1: Receive the cryptographic key $K_{IC} = K,\ K_S$ by means of a secure channel.
STEP 2: Receive the ciphertext block C using the unsecured channel.
STEP 3: As STEP 3 of Encryption.
STEP 4: As STEP 4 of Encryption.
STEP 5: Retrieve the plaintext block M from the ciphertext block C using the same routine `InvCipher` as presented in [1], but apply the value K_r as the actual parameter of the formal parameter Nr.

3. Conclusion

Although the implementation of the generalized AES algorithm requires some effort, this work is profitable, because it delivers not only a strong symmetric-key block cipher, but also a universal tool for exact examination of properties of algorithms AES-128, AES-192 and AES-256.

References

[1] **NIST**: *Advanced Encryption Standard (AES)*, FIPS PUB 197, 2001.

[2] **C. Kościelny**: *Computing in $GF(p^m)$ and in $gff(n^m)$ using Maple*, Quasigroups and Related Systems **13** (2005), 245 – 264.

[3] **A.J. Menezes, editor**: *Applications of Finite Fields*, Kluwer Academic Publishers, 1993.

Received July 13, 2005

Academy of Management in Legnica, Reymonta 21, 59-220 Legnica, Poland
e-mail: c.koscielny@wsm.edu.pl