Published in Fourth Conference on the AES - AES 2004, LNCS 3373

A Three Rounds Property of the AES

Marine Minier

Université Paris 8 - INRIA,
Projet CODES,
Domaine de Voluceau-Rocquencourt,
B.P. 105, 78 153 Le Chesnay Cedex - France
marine.minier@inria.fr

Abstract. Rijndael is the new Advanced Encryption Standard designed by V. Rijmen and J. Daemen and chosen as AES by the NIST in October 2000. Surprisingly, the number of cryptanalyses against this algorithm is very low in depict of many efforts furnished to break it.

This paper presents a stronger property than the one used in the Bottleneck Cryptanalysis [GM00]. Unfortunately, this property could not be used to mount a more efficient cryptanalysis than the Bottleneck Attack because it is not possible to improve the complexity of the four rounds distinguisher used in this attack. So, the complexity of the Bottleneck Attack (recalled in this paper) is always 2^{144} AES executions using 2^{32} plaintexts.

1 Introduction

In the initial article describing Rijndael [DR98], V. Rijmen and J. Daemen wrote : "For the different block lengths of Rijndael, no extensions to 7 rounds [of a known attack] faster than an exhaustive key search have been found". Of course, since 1998, some attacks reached this aim. In the case of key length equal to 192 or 256 bits, Ferguson et al., in [FKS+00], presented an improvement of the Square Attack [DR98] permitting to cryptanalyse an eight-rounds version of Rijndael with a complexity equal to 2^{204} executions and $2^{128} - 2^{119}$ plaintexts. S. Lucks presented in [Luc01] an other improvement of the Square Attack using a particular weakness of the key schedule against a seven-rounds version of Rijndael where 2^{194} executions are required for a number of chosen plaintexts equal to 2^{32}. H. Gilbert and M. Minier in [GM00] also presented an attack against a seven-rounds version of Rijndael (known under the name of "Bottleneck Attack") using a stronger property on three inner rounds than the one used in the Square Attack in order to mount an attack against a seven-rounds version of Rijndael requiring 2^{144} cipher executions with 2^{32} chosen plaintexts. In the case of a 128 bits key lenght, for a seven-rounds version of Rijndael, only two attacks are known. The first is due to Ferguson et al. in [FKS+00] and requires 2^{120} cipher executions for a number of plaintexts equal to $2^{128} - 2^{119}$. The second one, due to H. Gilbert and M. Minier in [GM00], is a marginal speed up of the 128-bits key search requiring 2^{32} chosen plaintexts.

H. Dobbertin, V. Rijmen, A. Sowa (Eds.): AES 2004, LNCS 3373, pp. 16–26, 2005.

Published in *Fourth Conference on the AES - AES 2004*, LNCS 3373

During the two last years, some new results was published concerning essentially the algebraic structure of the AES S-box. Those results use a potential weakness of the AES : there is only one non-linear operation in the AES round function, the inversion in the Galois Field $GF(2^8)$. So, it is possible to derive this inversion application into quadratic equations that are true with probability one. In [CP02], N. Courtois and J. Pieprzyk presents the quadratic equations given by the AES S-box on $GF(2)$. The authors use those quadratic equations to express all input/output bytes of each round and generate a huge system to solve. They apply the XL and the XSL algorithms to obtain the solutions of the system generated. An other article presented at Crypto'02 by S. Murphy and M. Robshaw [MR02] also describe the algebraic structure of the AES S-box and the quadratic equations of this one but on the field $GF(256)$.

The aim of this paper is to present a "new property" on three inner AES rounds stronger than the one used in the Bottleneck Attack described in [GM00]. This "new property" is very similar to the bottleneck property but as now does not permit to improve any attack due to the same number of dependent bytes implied in the four rounds distinguisher. This property is, however, stronger because the number of deduced collisions is bigger.

This paper is organized as follow: Section 2 provides a brief outline of the AES. Section 3 describes the 3-rounds property and the 4-rounds distinguisher used in the Bottleneck Attack. Section 4 presents the "new property". Section 5 describes, one more time, the bottleneck attack on seven rounds of the AES for a 128 bits block and a 192 or 256 bits key. Section 6 concludes this paper.

2 A Brief Outline of the AES

The AES is a symmetric block cipher using a parallel and byte-oriented structure. The key length and the block length are variable and are equal to 128, 192 or 256 bits. The current block is represented by a matrix of bytes. We focus from now on a 128-bits block represented by a 4×4 matrix of bytes :

$$B = \begin{array}{|c|c|c|c|} \hline b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ \hline b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\ \hline b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ \hline b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \\ \hline \end{array}$$

The number of rounds nr is also variable : 10, 12 or 14, depending on the block length and on the key length. The key schedule derives $nr + 1$ 128-bits round keys k_0 to k_{nr} from the master key k of variable length.

The round function, repeated $nr - 1$ times, is composed of four basic transformations, all linear except the first one :

- SubBytes : a bytewise transformation that applies on each byte of the current block an 8-bits to 8-bits non linear S-box (that we call S) composed by the inversion in the Galois Field $GF(256)$ and by an affine transformation.

Published in: *Fourth Conference on the AES, AES 2004*, LNCS 3373

- ShiftRows: a linear mapping that rotates on the left all the rows of the current matrix (0 for the first row, 1 for the second, 2 for the third and 3 for the fourth)
- MixColumn: another linear mapping represented by a 4×4 matrix chosen for its good properties of diffusion (see [DR02]). Each column of the input matrix is multiplied by the MixColumns matrix in the Galois Field $GF(256)$ that provides the corresponding column of the output matrix. We denote by $a_{i,j}$ for i and j from 0 to 3, the coefficients of the MixColumns matrix.
- AddRoundKey : a simple x-or operation between the input matrix and the subkey of the current round denoted by k_i.

Those $nr - 1$ rounds are surrounded at the top by an initial key addition with the subkey k_0 and at the bottom by a final transformation composed by a call to a round function where the MixColumns operation is omitted.

3 The Three-Rounds Property and the Four-Rounds Distinguisher

We now describe the three inner rounds property used in [GM00] and the four inner rounds distinguisher deduced.

3.1 The Three-Rounds Property

We note Y, Z, R and S the different intermediate input/output states of three consecutive inner rounds as noticed in figure 1.

We focus from now on an input block Y with its three left columns fixed. The most at right column, marked on figure 1, is composed by one active byte y which takes all possible values between 0 and 255 and by a triplet c equal to (c_0, c_1, c_2) of constant bytes which will represent a parameter. More formally, we note $\mathrm{Y}_{0,3} = y$, $\mathrm{Y}_{1,3} = c_0$, $\mathrm{Y}_{2,3} = c_1$ and $\mathrm{Y}_{3,3} = c_2$.

In the same way, we use the following notations for some particular bytes marked on figure 1. So, we denote $\mathrm{Z}_{0,3} = z_0$, $\mathrm{Z}_{1,3} = z_1$, $\mathrm{Z}_{2,3} = z_2$, $\mathrm{Z}_{3,3} = z_3$, $\mathrm{R}_{0,3} = r_0$, $\mathrm{R}_{1,0} = r_1$, $\mathrm{R}_{2,1} = r_2$, $\mathrm{R}_{3,2} = r_3$ and $\mathrm{S}_{0,3} = s$.

So, let us analyze how the Z, R and S particular bytes z_0 to z_3, r_0 to r_3 and s can be seen as c-dependent and key-dependent functions of the y input byte.

- After the first round, the $y \rightarrow z_0^c[y]$ one to one function is independent from the value of the c triplet and is entirely determined by one key byte, due to the effect of the ShiftRows operation. The same property holds for z_1, z_2 and z_3. So, the quartet of bytes (z_0, z_1, z_2, z_3) is a function of the y values entirely determined by four key-dependent bytes. More formally, there exists 4 key-dependent constants $k_{1,i,0}$ for $i = 0..3$ such that

$$z_i = a_{i,0} \cdot S(y) + k_{1,i,0} \ , \ i = 0..3$$

  where S represents the AES S-box.

Published in *Fourth Conference on the AES - AES 2004*, LNCS 3373

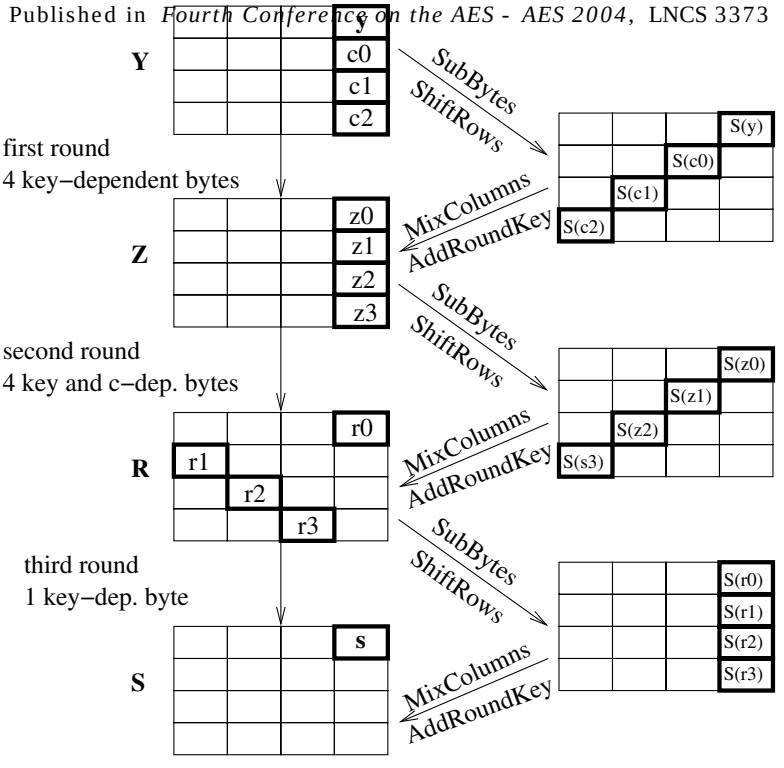

Fig. 1. The Three Inner Rounds Property

- After the second round, each of the four bytes $r_i[y]$, $i = 0..3$ is a one to one function of the corresponding $z_i[y]$ byte entirely determined by one single unknown byte that is entirely determined by c and the key. The quartet of bytes (r_0, r_1, r_2, r_3) of R marked on figure 1 is a function of (z_0, z_1, z_2, z_3) entirely determined by four key-dependent and c-dependent bytes.
- After the third round, the s byte marked on figure 1 can be expressed as a function of the (r_0, r_1, r_2, r_3) quartet of bytes entirely determined by one key-dependent byte depending on the subkey of the round.

In summary, the s byte depends on only 5 key-dependent bytes and 4 c-dependent bytes. More formally, the partial function $s^c[y]$ is entirely determined by a reduced number of unknown bytes. We can exploit this restricted dependency by constructing collisions on all the y values for distinct values of the c triplet. In other words :

Property 1. *There exists c' and c'' two triplets of constants such as for all y values between 0 and 255, we have : $s^{c'}[y] = s^{c''}[y]$. In this case, we say that we have a collision.*

Published in Fourth Conference on the AES - AES 2004 - LNCS 3373

In fact, the number of obtained collisions is 256, one for each y value.

Under the heuristic assumption that the unknown constants depending on the key and on the c triplet behave as random functions, then, by the birthday paradox, if we take a C set of 2^{16} c triplets, the probability to obtain a collision is non negligible.

This property can be extended to mount an efficient four-rounds distinguisher by adding a fourth round at the bottom of the three previous rounds.

3.2 The Four-Rounds Distinguisher

We consider the deciphering of the fourth round in the following way (see figure 2) :

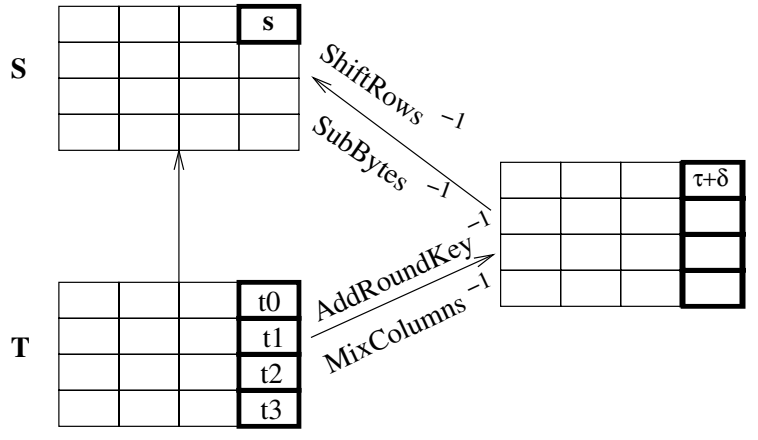

Fig. 2. The Extension to a Fourth Round

We denote by T the output block after the fourth round and we denote by (t_0, t_1, t_2, t_3) the last column of T marked on the figure 2. We can express the byte s as $s = S^{-1}[(0E \cdot t_0 + 0B \cdot t_1 + 0D \cdot t_2 + 09 \cdot t_3) + \delta]$ where S represents the AES S-box and δ a constant depending on the subkey of the fourth round. We have the following property :

Property 2. *There exists a collision between $s^{c'}[y]$ and $s^{c''}[y]$ if and only if for all y values between 0 and 255, we have :*

$$0E \cdot t_0^{c'} + 0B \cdot t_1^{c'} + 0D \cdot t_2^{c'} + 09 \cdot t_3^{c'} = 0E \cdot t_0^{c''} + 0B \cdot t_1^{c''} + 0D \cdot t_2^{c''} + 09 \cdot t_3^{c''}.$$

To simplify the notations we denote $0E \cdot t_0^c + 0B \cdot t_1^c + 0D \cdot t_2^c + 09 \cdot t_3^c$ by $\tau^c[y]$. $\tau^c[y]$ is a function of y entirely determined by 6 unknown bytes depending on the key and by 4 additional unknown bytes depending on both the key and the c values.

The following four inner rounds distinguisher is tested on a limited number of y values, a set Λ of 16 values is sufficient, the number of false alarms being negligible in this case.

Published in *Fourth Conference on the AES, AES 2004*, LNCS 3373

- Select a C set of about 2^{32} c triplet values and a subset of $\{0..255\}$, say for instance a Λ subset of 16 y values.
- For each c triplet value, compute the $L_c = (0E \cdot t_0^c + 0B \cdot t_1^c + 0D \cdot t_2^c + 09 \cdot t_3^c)_{y \in \Lambda}$.
- Check wether two of the above lists, $L_{c'}$ and $L_{c''}$ are equal.

The computations made at the secund step of this distinguisher (16 linear combinations of the outputs) represent substantially less than one single AES execution.

This four-rounds distinguisher requires about 2^{20} chosen inputs Y, and since the collision detection computations (based on the analysis of the corresponding T values) require less operations than the 2^{20} 4-inner rounds computations, the complexity of the distinguisher is less than 2^{20} AES encryptions for a probability of success equal to 1/2 (due to the birthday paradox).

4 The "New Three-Rounds Property"

We describe, in this section, a "new" three-rounds property derived from the previous one that permits to obtain an higher number of collisions. For more clarity, we use the same notation than the previous one. In this "new property", the number of initial active bytes in the last Y column has been modified. Here, we define two active bytes y and c_0 instead of one (the y byte) before. In this case, the c triplet becomes a pair of bytes defined by $c_p = (c_1, c_2)$. Let us explain how those two active bytes cross three inner rounds (see figure 3) :

- After the first round, the $y \rightarrow z_0^{c_p}[y, c_0]$ one to one function is independent from the value of the c_p triplet and is entirely determined by one key byte, due to the effect of the ShiftRows operation. The same property holds for z_1, z_2 and z_3. So, the quartet of bytes (z_0, z_1, z_2, z_3) is a function of the y and c_0 values entirely determined by four key-dependent bytes.
- After the second round, each of the four bytes $r_i[y, c_0]$, $i = 0..3$ is a one to one function of the corresponding $z_i[y, c_0]$ byte entirely determined by one single unknown byte that is entirely determined by c_p and the key. The quartet of bytes (r_0, r_1, r_2, r_3) of R marked on figure 3 is a function of (z_0, z_1, z_2, z_3) entirely determined by four key-dependent and c-dependent bytes.
- After the third round, the s byte marked on figure 3 can be expressed as a function of the (r_0, r_1, r_2, r_3) quartet of bytes entirely determined by one key-dependent byte depending on the subkey of the round.

As in the section 3.1, we can deduce that the s byte at the end of the third round is a function of y and c_0 entirely determined by 5 key-dependent bytes and 4 key-dependent and c_p-dependent bytes. We can also exploit this restricted dependency between the s byte and the two active bytes y and c_0 by defining a new kind of collision :

Published in *Fourth Conference on the AES - AES 2004*, LNCS 3373

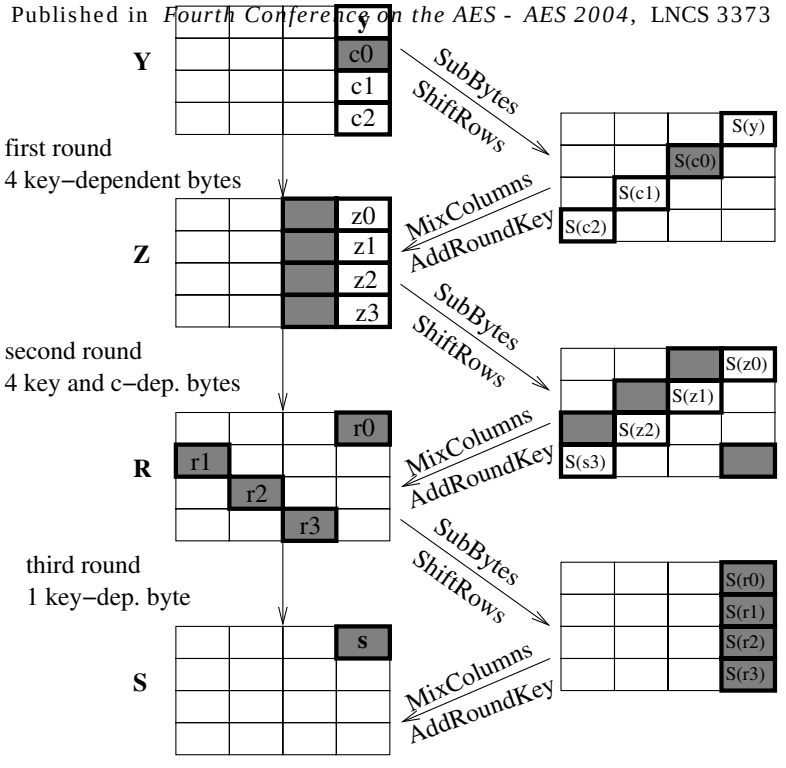

Fig. 3. The Other Three Inner Rounds Property

Property 3. *There exists c'_p and c''_p two pairs of constants such as for all y and c_0 values between 0 and 255, we have : $s^{c'_p}[y, c_0] = s^{c''_p}[y, c_0]$. In this case, we say that we have a collision.*

The number of obtained collisions is $(256)^2$, one for each y and c_0 value.

This "new" property is due to the very symmetric and parallel structure of the AES in the byte position level.

We verify the veracity of this property by computer experiments.

Unfortunately, this property could not be used to mount a more efficient four-rounds distinguisher than the one presented in section 3.2. Indeed, the number of c_p pairs used in the distinguisher and the probability of success depend on only the four intermediate c_p-dependent bytes. The number of such bytes is the same for the property of the section 3.1 and the "new" one in depict of the bigger number of obtained collisions.

So, we do not find a more efficient distinguisher that permits to use this stronger property and to improve an attack. We use the same distinguisher than the one described in section 3.2.

Published in: Fourth Conference on the Advanced Encryption Standard - AES 2004, LNCS 3373

5 The Bottleneck Attack on a Seven-Rounds Version of the AES with Key Lengths Equal to 192 and 256 Bits Using 2^{32} Chosen Plaintexts

Even if the new property could not be used to improve the four-rounds distinguisher described in section 3.2 and so the bottleneck attack, we give a short description of this known cryptanalysis. So, we are going to recall, in this section, how the four inner rounds distinguisher of the section 3.2 could be extended to mount a seven-rounds attack on the AES with a 192 or a 256 bits key.

The seven-rounds version of the AES is depicted in figure 4. The seventh round is here considered as the last round (i.e. it doesn't contain the MixColumns operation). X represents a plaintext block and V the corresponding ciphertext. The four previous rounds are surrounded at the top by one initial round X $\rightarrow$ Y, composed by an initial key addition followed by one round and at the bottom by two final rounds : T $\rightarrow$ U and U $\rightarrow$ V.

The attack method is a combination between the four-rounds distinguisher presented in section 3.2 and an exhaustive search of some keybytes or combination of keybytes of the initial and the two final rounds. The attack described here uses the fact that, in the equations provided by the four-rounds distinguisher, there is a variables separation in terms which involve one half of the 2 last rounds key bytes and terms which involve a second half of the 2 last round key bytes in order to save a 2^{80} factor in the exhaustive search complexity.

5.1 Extension at the Beginning

The distinguisher of section 3.2 could be extended by one round at the beginning using the same method than the one proposed by the authors of Rijndael in the initial paper [DR98] and first applied to the algorithm Square.

The main idea used here is that if, in the initial key addition, the 4 key bytes (denoted by $k_{ini} = (k_{0,0}, k_{0,1}, k_{0,2}, k_{0,3})$), added with the four bytes (x_0, x_1, x_2, x_3) of the plaintext X marked in figure 3, are known then it is possible to partition the 2^{32} plaintexts into 2^{24} subsets of 2^8 plaintexts values satisfying the conditions of section 3.2 (i.e. (c_0, c_1, c_2) stay a triplet of constants and y is the active byte).

So, if all the 2^{32} possible plaintexts are encrypted for all the possible values of the (x_0, x_1, x_2, x_3) quartet (the other 12 bytes being taken equal to a constant), the 2^{32} plaintexts could be partitioned, according to the value of k_{ini}, into 2^{24} subsets of 2^8 plaintexts according the values of y (which are known up to an unknown constant linked with the first round key byte). Those subsets are such that the y byte takes all possible values between 0 and 255 and the $c = (c_0, c_1, c_2)$ triplet is composed of three constant values, different and unique for each of the 2^{24} subsets, the 12 other Y bytes are constant and all those constant values are the same for all subsets.

Those 2^{32} plaintexts give the corresponding 2^{32} ciphertexts V.

Published in *Fourth Conference on the AES - AES 2004*, LNCS 3373

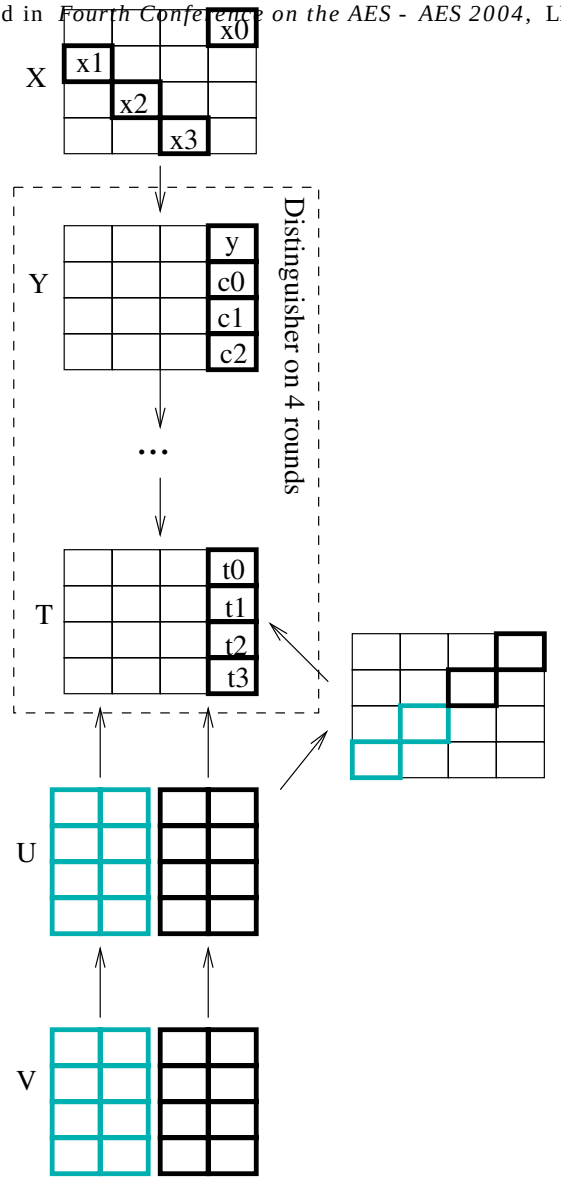

Fig. 4. Attack on Seven Rounds of the AES

5.2 Extension at the End

Each of the t_0, t_1, t_2, t_3 bytes can be expressed as a function of four bytes of the V ciphertext and five unknown key bytes (i.e. four of the final round subkey and one linear combination of the penultimate round subkey). So, in order to improve the key exhaustive search on the two last rounds, the equations of collisions are "cuted" in two parts as follows :

Published in: For the Conference on the AES - AES 2000 - LNCS 3373

With this notation the equation of collision $\tau^{c'} = \tau^{c''}$ described in property 2 could be expressed as $\tau_1^{c'} + \tau_2^{c'} = \tau_1^{c''} + \tau_2^{c''}$, i.e. $\tau_1^{c'} + \tau_1^{c''} = \tau_2^{c'} + \tau_2^{c''}$. τ_1 depends on t_0 and t_1 and τ_2 depends on t_2 and t_3. Now, due to the fact that the last round of the AES does not contain the MixColumns operation, t_0 and t_1 could be expressed, as shown on figure 3, as a function of 8 ciphertext bytes and 10 key bytes of the two last rounds denoted by k_{τ_1}. In the same way, t_2 and t_3 depend on 8 ciphertext bytes and on 10 key bytes of the two last rounds denoted by k_{τ_2}.

This remark permits to share in two parts the key exhaustive search and to improve the attack on a seven rounds-version of the AES by a factor 2^{80}.

5.3 Outline of the Attack

An efficient exhaustive search of the k_{ini}, k_{τ_1} and k_{τ_2} keys could be performed in the following way:

First step :

Cipher the 2^{32} chosen plaintexts for all possible values of the quartet (x_0, x_1, x_2, x_3).

Second step :

For k_{ini} from (0,0,0,0) to (255,255,255,255) do
  Partition the $(256)^4$ chosen plaintexts
  into $(256)^3$ Λ^c sets according the value of the triplet c
  Choose into those $(256)^3$ Λ^c sets 2^{16} values of c
  For each value of the (c', c'') pair do
    For k_{τ_1} from $(0, \cdots, 0)$ to $(255, \cdots, 255)$ do
      Compute the values of $(\tau_1^{c'} \oplus \tau_1^{c''})_{y=0\cdots15}$ from the ciphertexts
      Put them in a table $T_{k_{ini},c',c''}[k_{\tau_1}]$
    End For
    For k_{τ_2} from $(0, \cdots, 0)$ to $(255, \cdots, 255)$ do
      Compute the values of $(\tau_2^{c'} \oplus \tau_2^{c''})_{y=0\cdots15}$ from the ciphertexts
      Look in the table $T_{k_{ini},c',c''}[k_{\tau_1}]$ if the same values appear
      If yes, verify the same computation for all the y values
        If equality for all y values, return $(k_{ini}, k_{\tau_1}, k_{\tau_2})$
      Else continue
      End If
    End For
  End For
End For

Since the above procedure tests whether the exist collisions inside a random set of 256^2 of the 256^4 possible $s^c[y]$ functions, the probability of the procedure to result in a collision, and thus to provide k_{ini}, k_{τ_1} and k_{τ_2} is high (say about 1/2). In other words, the success probability of the attack is about 1/2.

The first step could be made independently and requieres 2^{32} chosen plaintexts and 2^{32} AES executions.

Published in *Fourth Conference on the AES - AES 2004*, LNCS 3373

The complexity of the second step is about 2^{25} operations, less expensive than AES executions. Its probability of success is about 1/2. This attack provides 20 bytes of information on the last and penultimate key values.

5.4 How to Improve this Attack Using the Lucks' Property of the Key Schedule for a 192 Bits Key

We can improve, by using the particular property of the key schedule described by S. Lucks in [Luc00], the complexity of the attack by a little factor in the case of a key length equal to 192 bits. Indeed, the attack presented by S. Lucks permits to limit the key exhaustive search to only 8 k_{τ_2} bytes instead of the 10 initial bytes because the knowledge of the two first columns of the last subkey determines completely, taking into account the effect of the last ShiftRow, the two others bytes of the penultimate subkey that compose k_{τ_2}.

6 Conclusion

We have shown in this paper that there exists a strong collision property on three inner AES rounds due to some partial byte oriented functions induced by the AES cipher. This property is stronger than the one used in the bottleneck attack even if this new bottleneck property could not be extend in a better four rounds distinguisher that the one used in the known attack.

Maybe, there is a better way to exploit this new restricted dependency but we do not find how to extend it.

References

[AES99] http://www.nist.gov/aes

[CP02] N. Courtois and J. Pieprzyk, "Cryptanalysis of Block Ciphers with Overdefined Systems of Equations". In *Asiacrypt'02*, Queenstown, New Zealand, Lecture Notes in Computer Science 2501, Springer-Verlag, 2002.

[DR98] J. Daemen, V. Rijmen, "AES Proposal : Rijndael", *The First Advanced Encryption Standard Candidate Conference,* N.I.S.T., 1998.

[DR02] J. Daemen, V. Rijmen, *The Design of Rijndael*. Springer-Verlag, 2002.

[FKS+00] N. Ferguson, J. Kelsey, B. Shneier, M. Stay, D. Wagner and D. Whiting, "Improved Cryptanalysis of Rijndael". In *Fast Software Encryption'00*, New York, United State, pp. 213-230. Lectures Notes in Computer Science 1978, Springer-Verlag, 2000.

[GM00] H. Gilbert, M. Minier, "A Collision Attack on 7 rounds of Rijndael". In *The Third Advanced Encryption Standard Candidate Conference*. N.I.S.T., 2000.

[Luc00] S. Lucks, "Attackng Seven Rounds of Rijndael Under 192-bit and 256-bit Keys". In *The Third Advanced Encryption Standard Candidate Conference*. N.I.S.T., 2000.

[MR02] S. Murphy and M. Robshaw, "Essential Algebraic Structure Within the AES". In *Crypto'02*, Santa Barbara, United State, Lectures Note in Computer Science 2442, Springer-Verlag, 2002.