Using AES Round Symmetries to Distinguish PAES

Jérémy Jean and Ivica Nikolić

Nanyang Technological University, Singapore
{JJean, INikolic}@ntu.edu.sg

Abstract. We show that PAES has a class of 2^{64} weak keys (out of 2^{128}) that can be used to distinguish the ciphertext from a random in a chosen-nonce attack framework. The distinguisher requires only one block of ciphertext (the very first one), and has a negligible time complexity. In a nutshell, it exploits the lack of constants in PAES and the symmetric properties of the keyless AES round function.

Keywords: PAES, authenticated encryption, distinguisher

1 Introduction

PAES [3] is an authenticated encryption scheme proposed for the CAESAR competition [1]. In this note we show that not all 2^{128} key of PAES are secure for use – rather there is a class of 2^{64} weak keys. If PAES is instantiated with such key, and the adversary can choose the nonce, then the cipertext stream can be distinguished from a random one with only a single query to the encryption oracle and negligible complexity. This attack framework is specified in the submission document (see Attacking goals, point 3, on page 8 of [3]).

The security goals of the CAESAR competition are not clear against attacks that use weak keys. We consider that these goals should meet the similar security levels as in the case of more traditional block/stream ciphers – there, a weak key attack that uses one plaintext (and a few operations) to distinguish the ciphertext from random, is considered a valid attack.

In the distinguisher presented further, we exploit the following two facts:

- The initialization of PAES and all the calls to AES rounds do not use any constants
- The AES round function perserves certain symmetries if composed of only SubBytes, ShiftRows, MixColumns (i.e., no AddRoundKeys, or with an all-zero subkey)

We describe further the distinguisher for PAES-4, and we note that it can be applied to PAES-8 with minor modifications.

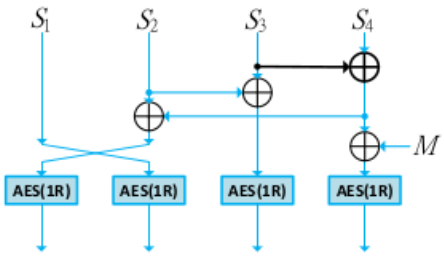

Fig. 1. The round function $StateUpdate(S, M)$. During the processing of the plaintext, the xor from S_3 to S_4 is absent. The image is taken from [3].

2 Specifications

We give minimal description of PAES-4 which should be sufficient to understand the attack. The design resembles a stream cipher: it has an initialization (where the key and the nonce are loaded into the state), then it process the input message and produces ciphertexts, and finally in the finalization it produces the tag. The internal state S has 4 words S_1, S_2, S_3, S_4, each of 128 bits, i.e. $|S_i| = 128, i = 1, 2, 3, 4$. The state update function $StateUpdate(S, M)$ is the round transformation and uses 4 AES-round calls to update the state (see Figure 1).

We emphasize that all the AES calls are keyless, that is, composed of SubBytes, ShiftRows and MixColumns (but no AddRoundKey).

Initialization. The 128-bit master key K and the nonce N are loaded into the four words of the state, the state goes through 5 rounds and at the end the key is xored to all four words of the state:

$$
\begin{aligned}
&S_1 = K \oplus N \\
&S_2 = L(K) \oplus L^3(N) \\
&S_3 = L^2(K) \oplus L(N) \\
&S_4 = L^3(K) \oplus L^2(N) \\
&for\ i = 1\ to\ 5 \\
&\quad S = StateUpdate(State, 0) \\
&for\ i = 1\ to\ 4 \\
&\quad S_i = S_i \oplus K
\end{aligned}
$$

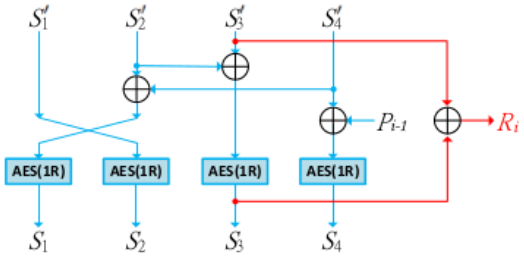

Fig. 2. One round of encryption. The image is taken from [3].

where L is the linear transformation that operates on the four 32-bit columns a, b, c, d of a 128-bit word $a||b||c||d$, and is defined as $L(a, b, c, d) = (b, c, d \oplus a, a)$.

Processing the plaintext. In one round, from 16-byte plaintext P_i, 16-byte ciphertext C_i is obtained with one call to the $StateUpdate$ (see Fig 2):

$$
\begin{aligned}
&tmp = S_3\\
&StateUpdate(S, P_i)\\
&R_i = tmp \oplus S_3\\
&C_i = P_i \oplus R_i
\end{aligned}
$$

3 Symmetric Properties of the AES

We specify here how the known symmetric property [2] of the AES applies in the case of PAES. Namely, if an state is symmetric in the sense that its two halves are equal, then the keyless round function of the AES maintains this property. In the sequel, we denote $\texttt{AES}_0$ the keyless AES round function. We recall the property of [2] using block matrices, and we introduce the more general following notations:

$$U(A, B) = \left(\begin{array}{c|c} A & A \\ \hline B & B \end{array}\right), \qquad V(A, B) = \left(\begin{array}{c|c} A & B \\ \hline B & A \end{array}\right), \qquad W(A, B) = \left(\begin{array}{c|c} A & B \\ \hline A & B \end{array}\right).$$

Additionally, we denote $\mathcal{U}$, $\mathcal{V}$ and $\mathcal{W}$ the associated sets respectively for all possible values of the 2×2 block matrices A and B. Finally, we denote M the constant MDS matrix used in the AES round function, and observe

that:

$$M = \begin{pmatrix} 2\,3\,1\,1 \\ 1\,2\,3\,1 \\ 1\,1\,2\,3 \\ 3\,1\,1\,2 \end{pmatrix} = \left(\begin{array}{c|c} M_1 & M_2 \\ \hline M_2 & M_1 \end{array}\right) = V(M_1, M_2) \in \mathcal{V}.$$

Property 1. Let $S \in \mathcal{U}$. Then, $\mathtt{AES}_0(S) \in \mathcal{U}$.

Proof. Let $S = U(A, B) \in \mathcal{U}$, and write the bytes in S as:

$$\left(\begin{array}{c|c} A & A \\ \hline B & B \end{array}\right) = \left(\begin{array}{cc|cc} x_0 & x_4 & x_0 & x_4 \\ x_1 & x_5 & x_1 & x_5 \\ \hline x_2 & x_6 & x_2 & x_6 \\ x_3 & x_7 & x_3 & x_7 \end{array}\right).$$

As the SubBytes operation applies the same bijection to all the bytes in the state, we ignore it here as it obviously preserves the structure. However, after the ShiftRows operation, the state becomes

$$\left(\begin{array}{cc|cc} x_0 & x_4 & x_0 & x_4 \\ x_5 & x_1 & x_5 & x_1 \\ \hline x_2 & x_6 & x_2 & x_6 \\ x_7 & x_3 & x_7 & x_3 \end{array}\right) \stackrel{\text{def}}{=} \left(\begin{array}{c|c} A' & A' \\ \hline B' & B' \end{array}\right)$$

which still belongs to $\mathcal{U}$. Then, the MixColumns operation gives:

$$\begin{aligned} \begin{pmatrix} 2\,3\,1\,3 \\ 1\,2\,3\,1 \\ 1\,1\,2\,3 \\ 3\,1\,1\,2 \end{pmatrix} \times \begin{pmatrix} x_0\ x_4\ x_0\ x_4 \\ x_5\ x_1\ x_5\ x_0 \\ x_2\ x_6\ x_2\ x_6 \\ x_7\ x_3\ x_7\ x_3 \end{pmatrix} &= \left(\begin{array}{c|c} M_1 & M_2 \\ \hline M_2 & M_1 \end{array}\right) \times \left(\begin{array}{c|c} A' & A' \\ \hline B' & B' \end{array}\right) \\ &= \left(\begin{array}{c|c} M_1A' \oplus M_2B' & M_1A' \oplus M_2B' \\ \hline M_2A' \oplus M_1B' & M_2A' \oplus M_1B' \end{array}\right) \\ &\stackrel{\text{def}}{=} \left(\begin{array}{c|c} A'' & A'' \\ \hline B'' & B'' \end{array}\right) \in \mathcal{U}. \end{aligned}$$

□

Property 2. Let $S \in \mathcal{W}$. Then, $\mathtt{AES}_0(S) \in \mathcal{V}$, and $\mathtt{AES}_0(\mathtt{AES}_0(S)) \in \mathcal{W}$.

Proof. Let $S = W(A, B) \in \mathcal{W}$, and write the bytes in S as:

$$\left(\begin{array}{c|c} A & B \\ \hline A & B \end{array}\right) = \left(\begin{array}{cc|cc} x_0 & x_2 & x_4 & x_6 \\ x_1 & x_3 & x_5 & x_7 \\ \hline x_0 & x_2 & x_4 & x_6 \\ x_1 & x_3 & x_5 & x_7 \end{array}\right).$$

Again, we ignore the SubBytes operation as the bijection applied preserves the structure of the internal states. However, after the ShiftRows operation, the state becomes:

$$\left(\begin{array}{cc|cc} x_0 & x_2 & x_4 & x_6 \\ x_3 & x_5 & x_7 & x_1 \\ \hline x_4 & x_6 & x_0 & x_2 \\ x_7 & x_1 & x_3 & x_5 \end{array}\right) \stackrel{\text{def}}{=} \left(\begin{array}{c|c} A' & B' \\ \hline B' & A' \end{array}\right) \in \mathcal{V},$$

which is tranformed by the subsequent MixColumns tranformation into the state:

$$\begin{aligned} \left(\begin{array}{c|c} M_1 & M_2 \\ \hline M_2 & M_1 \end{array}\right) \times \left(\begin{array}{c|c} A' & B' \\ \hline B' & A' \end{array}\right) &= \left(\begin{array}{c|c} M_1A' \oplus M_2B' & M_1B' \oplus M_2A' \\ \hline M_2A' \oplus M_1B' & M_2B' \oplus M_1A' \end{array}\right) \\ &\stackrel{\text{def}}{=} \left(\begin{array}{c|c} A'' & B'' \\ \hline B'' & A'' \end{array}\right) \in \mathcal{V}. \end{aligned}$$

After applying a second keyless `AES` round, we get:

$$\left(\begin{array}{c|c} A'' & B'' \\ \hline B'' & A'' \end{array}\right) = \left(\begin{array}{cc|cc} y_0 & y_2 & y_4 & y_6 \\ y_1 & y_3 & y_5 & y_7 \\ \hline y_4 & y_6 & y_0 & y_2 \\ y_5 & y_7 & y_1 & y_3 \end{array}\right) \xrightarrow{\mathsf{SR}} \left(\begin{array}{cc|cc} y_0 & y_2 & y_4 & y_6 \\ y_3 & y_5 & y_7 & y_1 \\ \hline y_0 & y_2 & y_4 & y_6 \\ y_3 & y_5 & y_7 & y_1 \end{array}\right) \stackrel{\text{def}}{=} \left(\begin{array}{c|c} A''' & B''' \\ \hline A''' & B''' \end{array}\right) \in \mathcal{W},$$

and by the MixColumns:

$$\begin{aligned} \left(\begin{array}{c|c} M_1 & M_2 \\ \hline M_2 & M_1 \end{array}\right) \times \left(\begin{array}{c|c} A''' & B''' \\ \hline A''' & B''' \end{array}\right) &= \left(\begin{array}{c|c} M_1A''' \oplus M_2A''' & M_1B''' \oplus M_2B''' \\ \hline M_2A''' \oplus M_1A''' & M_2B''' \oplus M_1B''' \end{array}\right) \\ &\stackrel{\text{def}}{=} \left(\begin{array}{c|c} A'''' & B'''' \\ \hline A'''' & B'''' \end{array}\right) \in \mathcal{W}, \end{aligned}$$

which concludes the proof. □

Finally, we can represent the action of the keyless `AES` round function $\texttt{AES}_0$ of the three sets $\mathcal{U}$, $\mathcal{V}$ and $\mathcal{W}$ as follows on Figure 3.

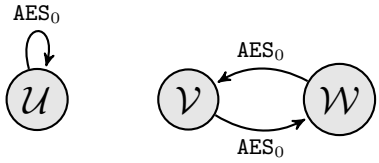

Fig. 3. Action of $\texttt{AES}_0$ of the symmetrical states from $\mathcal{U}$, $\mathcal{V}$ and $\mathcal{W}$.

4 The Distinguisher

To distinguish PAES, we use the first ciphertext C_1 produced during the encryption of the plaintext $P_1 = 0^{128}$ with a secret key $K \in \mathcal{W}$ and nonce $N \in \mathcal{W}$. The key K can be any of such 2^{64} keys (the first two rows equal to the second two rows), and a similar holds for the nonce N.

Let us inspect first how the state words S_1, S_2, S_3, S_4 change the class belongings (either $\mathcal{W}$ or $\mathcal{V}$) from the very first to the last steps of the Initialization:

- $K, N \in \mathcal{W}$. For any input $X \in \mathcal{W}$, the output of the linear function $L(X) \in \mathcal{W}$, that is, if the input is in $\mathcal{W}$, then L does not change the class. Therefore, $S_1, S_2, S_3, S_4 \in \mathcal{W}$ after the initial assignments in the initalization.
- After the first update. The xors do not change the class belongings, thus each S_i stays in $\mathcal{W}$ after the xors at the top of the *StateUpdate*. Further, according to the properties presented in the previous section, the AES rounds change the class from $\mathcal{W}$ to $\mathcal{V}$, thus at the end of the first update, $S_i \in \mathcal{V}$.
- After the second update. Similarly as for the previous, but this time the class of S_i changes to $\mathcal{W}$.
- After the third update. The class of S_i are all $\mathcal{V}$.
- After the fourth update. The class of S_i are all $\mathcal{W}$.
- After the fifth update. The class of S_i are all $\mathcal{V}$.
- After the xors of the key. Each S_i has the form $X_i \oplus K$, where $X_i \in \mathcal{V}$, and $K \in \mathcal{W}$.

Now lets focus on the production of the ciphertext C_1. We can see that at the top of the *StateUpdate*, first S_2 is xored to S_3. The resulting word Y_3 it must belongs then to $\mathcal{V}$ since $X_2 \oplus K \oplus X_3 \oplus K = X_2 \oplus X_3 \in \mathcal{V}$. The appliction of the AES round to Y_3 results in a word Z_3 from the class $\mathcal{W}$.

The ciphertext C_1 is defined as

$$C_1 = (X_3 \oplus K) \oplus Z_3 \oplus M_1 = X_3 \oplus K \oplus Z_3,$$

where $X_3 \in \mathcal{V}, K \in \mathcal{W}, Z_3 \in \mathcal{W}$. Let,

$$X_3 = \left(\begin{array}{c|c} X^A & X^B \\ \hline X^B & X^A \end{array}\right), K = \left(\begin{array}{c|c} K^A & K^B \\ \hline K^A & K^B \end{array}\right), Z_3 = \left(\begin{array}{c|c} Z^A & Z^B \\ \hline Z^A & Z^B \end{array}\right),$$

then

$$C_1 = \left(\begin{array}{c|c} C^A & C^B \\ \hline C^C & C^D \end{array}\right) = \left(\begin{array}{c|c} X^A \oplus K^A \oplus Z^A & X^B \oplus K^B \oplus Z^B \\ \hline X^B \oplus K^A \oplus Z^A & X^A \oplus K^B \oplus Z^B \end{array}\right).$$

Obviously $C^A \oplus C^B \oplus C^C \oplus C^D = 0$, hence the xor of the four 32-bit blocks of the first ciphertext must result in a zero block.

As a result, we have a distinguisher which requires negligible complexity and only one block of plaintext/ciphertexts to distinguish `PAES` when instantiated with any of the 2^{64} keys and nonces from the class $\mathcal{W}$.

We note that our computer simulation confirmed the correctness of the distinguisher.

5 Conclusion

In this paper, we have shown a distinguisher based on the `AES` round symmetry used as an underlying transformation in the authenticated encryption scheme `PAES`. The distinguisher works for a fraction of 2^{64} out of the total 2^{128} keys in the framework of chosen-nonce attacks, requires only a single block of ciphertexts and has negligible complexity.

The distinguisher can be stopped easily with the use of keyed `AES` rounds under randomly chosen constants as keys. Using such `AES` rounds even only in the initialization should prevent the exploitation of the `AES` round symmetry in the later rounds of the encryption. In addition, using non-zero message inputs in the initalization could result in a design resistant against our distinguisher.

We note that `PAES`-8 can be attacked with the very same approach as for `PAES`-4.

References

1. CAESAR Competition. `http://competitions.cr.yp.to/caesar.html`.
2. T. V. Le, R. Sparr, R. Wernsdorf, and Y. Desmedt. Complementation-Like and Cyclic Properties of AES Round Functions. In H. Dobbertin, V. Rijmen, and A. Sowa, editors, *AES Conference*, volume 3373 of *Lecture Notes in Computer Science*, pages 128–141. Springer, 2004.
3. D. Ye, P. Wang, L. Hu, L. Wang, Y. Xie, S. Sun, and P. Wang. PAES v1: Parallelizable Authenticated Encryption Schemes based on AES Round Function. Submission to the CAESAR competition, 2014, 2014.