Linear Biases in AEGIS Keystream

Brice Minaud

ANSSI, 51, boulevard de la Tour-Maubourg, 75700 Paris 07 SP, France
brice.minaud@gmail.com

Abstract. AEGIS is an authenticated cipher introduced at SAC 2013, which takes advantage of AES-NI instructions to reach outstanding speed in software. Like LEX, Fides, as well as many sponge-based designs, AEGIS leaks part of its inner state each round to form a keystream. In this paper, we investigate the existence of linear biases in this keystream. Our main result is a linear mask with bias 2^{-89} on the AEGIS-256 keystream. The resulting distinguisher can be exploited to recover bits of a partially known message encrypted 2^{188} times, regardless of the keys used. We also consider AEGIS-128, and find a surprising correlation between ciphertexts at rounds i and $i+2$, although the biases would require 2^{140} data to be detected. Due to their data requirements, neither attack threatens the practical security of the cipher.

Key words: Cryptanalysis, AEGIS, CAESAR

1 Introduction

Traditional block cipher-based encryption ensures the confidentiality of encrypted data: it is infeasible for anyone to decipher a message without knowledge of the secret encryption key. However there is a compelling need for ciphers achieving at once confidentiality and authenticity; that is, ciphers integrating a form of integrity check guaranteeing that the encrypted message does originate from its purported sender. Any tampering of the data will result in its rejection by the deciphering algorithm. The CAESAR [cae13] authenticated cipher competition, sponsored by the National Institute of Standards and Technology, crystallizes the community's growing interest in this type of cipher. In March 2014, first round submissions were finalized and all entries were published online, awaiting analysis.

AEGIS [WP14] is a particularly notable candidate in this competition. Indeed, it takes full advantage of the new AES-NI instruction set in recent Intel and AMD processors to achieve unprecedented encryption speed in software, at around half a cycle per byte. Although AEGIS was first introduced only a year ago at SAC 2013, it has already inspired other encryption designs, including PAES [YWH+14] and Tiaoxin [Nik14]. The state update function at the core of AEGIS is simply the parallel application of a single AES round to a large state, followed by a shift and `XOR`. This exploits the pipeline implementation of AES-NI, which allows for the parallel computation of several AES rounds.

AEGIS, like many entries in the CAESAR competition, follows a model where a large inner state leaks essentially a portion of itself every round, which is then `XOR`-ed with the plaintext to form the ciphertext. Moreover, like most ciphers in this family, including all duplex-like constructions [BDPA12], it delays the insertion of a plaintext block into the inner state until after the corresponding ciphertext block has been output, in order for decryption to proceed in the same direction as encryption. As such, these ciphers are not proper stream ciphers, but form an interesting hybrid, where a single round behaves like a stream cipher.

In particular, assume we have a linear distinguisher on the ciphertext with known plaintext, which we call a keystream bias by analogy with stream ciphers. That is, we know that the sum of some specific bits of the ciphertext is biased towards 0 or 1, provided the corresponding plaintext has a known value. Then, because of the stream cipher-like behavior pointed out above, if only the last block of plaintext involved varies, and the rest remains fixed as before, the sum of ciphertext bits is biased towards 0 or 1 depending on the same sum on the plaintext.

Thus, a linear distinguisher on the keystream yields an attack on the scheme, where plaintext bits of a partially known message can be recovered, provided the message is encrypted enough times. Observe

that this does not require the same key be used. Indeed, this plaintext could be encrypted in entirely different sessions with different keys, as long as it is encrypted a sufficient number of times in total. This is very reminiscent of classic stream cipher attacks such as linear masking [CHJ02], as well as recent attacks on RC4 [ABP+13]. However, in the security analysis of AEGIS by its authors, as well as many CAESAR submissions displaying similar stream cipher-like behavior, this type of attacks does not seem to be taken into account. This leaves open the question of how effective they might be, which we investigate for AEGIS.

Our contribution.

In this paper, we describe linear biases in the keystream of AEGIS-128 and AEGIS-256. As far as we know, this is the first cryptanalysis of AEGIS. These biases result from the surprising property that, although the inner state of AEGIS-128 (resp. AEGIS-256) is 5 (resp. 6) times the size of its output per round, the outputs of only 3 consecutive rounds are related. This is particularly striking in the case of AEGIS-128, where we show that the outputs of rounds i and $i+2$ are correlated.

However, the biases we find are quite small. In the case of AEGIS-256, we exhibit biases of 2^{-89} for a few linear masks, which would require 2^{188} data to be detected with good probability. This bias only requires a known plaintext to be encrypted repeatedly, with no assumption about the keys or nonces: in fact, the inner state before encryption is considered uniformly random. This distinguisher can also be exploited to recover information on a partially known plaintext encrypted 2^{188} times. Due to the data requirements involved, our attack does not threaten the practical security of the cipher. For instance, restricting the attacker to not use more than 2^{128} data in total even for AEGIS-256, independently of the keys involved, would most likely prevent this type of attack entirely.

We also investigate linear biases of AEGIS-128, and find a bias of 2^{-77} between outputs of the cipher at rounds i and $i+2$. While this would require more than 2^{128} data, it is still worth noting, as this bias is vastly superior to any generic attack, considering the inner state is 640-bit long. In Appendix B, we investigate to what extent linear hull effects as well as multilinear techniques can be expected to reduce the data requirements. We find that around 2^{140} data would likely still be required, showing that AEGIS-128 should be safe from our attack.

The first section provides a brief description of AEGIS-128 and AEGIS-256 encryption. In the second sction, we define linear biases, and study some linear biases linking substates of AEGIS. From there, we deduce biases in the keystream of AEGIS-128 and AEGIS-256. Finally, we show how these biases can be exploited to mount an attack.

1.1 Notations

For:	n	an integer
	X	a n-bit vector
	Y	a n-bit vector
	α	a n-bit vector
Define:	$X \oplus Y$	the bitwise `XOR` of X and Y
	$X \& Y$	the bitwise `AND` of X and Y
	$\alpha \cdot X$	the scalar product of α and X
	$\lvert\alpha\rvert$	the Hamming weight of α

2 Description of AEGIS-128 and AEGIS-256

When AEGIS was first introduced at SAC 2013, it came in two variants, AEGIS-128 and AEGIS-256, providing a security level of 128 and 256 bits respectively. In the CAESAR proposal, a new variant is introduced, AEGIS-128L, which fully leverages the 8-stage AES pipeline provided by Intel Sandy Bridge processors. In this paper, we focus on AEGIS-128 and AEGIS-256 in their most recent version, namely the CAESAR submission [cae13].

2.1 AEGIS-128

AEGIS-128 takes as parameters a 128-bit key, a 128-bit nonce, and a tag length less than or equal to 128. It proceeds in several stages: initialization, where the 640-bit inner state is initialized using the key and nonce; processing of the authenticated data, where optional associated data is integrated into the state; encryption proper, where a variable-length plaintext is encrypted into a ciphertext of the same length; and finalization, which produces an authentication tag from the inner state. Hereafter we are only interested in the encryption step. A complete description of AEGIS can be found in [WP14].

The inner state of AEGIS-128 consists of five 128-bit substates $S_0, \ldots, S_4$. The plaintext is divided into 128-bit blocks m_i, $i \geq 0$, and processed in successive rounds. Let us denote by $S_{i,0}, \ldots, S_{i,4}$ the values of the substates at round i. For simplicity, we set $i = 0$ when encryption begins, setting aside the initialization step as well as the processing of authenticated data.

Then we have:

$$\begin{aligned}
S_{i+1,0} &= S_{i,0} \oplus \mathrm{R}(S_{i,4}) \oplus m_i \\
S_{i+1,1} &= S_{i,1} \oplus \mathrm{R}(S_{i,0}) \\
S_{i+1,2} &= S_{i,2} \oplus \mathrm{R}(S_{i,1}) \\
S_{i+1,3} &= S_{i,3} \oplus \mathrm{R}(S_{i,2}) \\
S_{i+1,4} &= S_{i,4} \oplus \mathrm{R}(S_{i,3})
\end{aligned}$$

where R denotes a single round of AES-128 [DR99], with no key addition. The state update function is depicted in Figure 1.

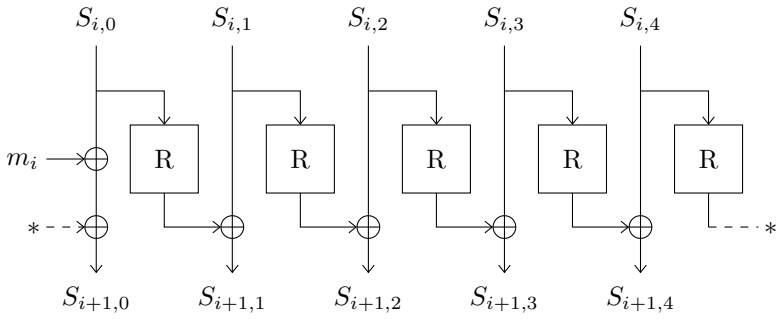

Fig. 1. State update function of AEGIS-128.

Each round, the ciphertext is output as:

$$C_i = S_{i,1} \oplus S_{i,4} \oplus (S_{i,2} \,\&\, S_{i,3}) \oplus m_i$$

2.2 AEGIS-256

AEGIS-256 takes as parameters a 256-bit key, a 256-bit nonce, and a tag length less than or equal to 128. The encryption step is very much the same as that of AEGIS-128, except the inner state consists of six rather than five 128-bit substates $S_{i,0}, \ldots, S_{i,5}$. The state update function may be written as:

$$\begin{aligned}
S_{i+1,0} &= S_{i,0} \oplus \mathrm{R}(S_{i,5}) \oplus m_i \\
S_{i+1,1} &= S_{i,1} \oplus \mathrm{R}(S_{i,0}) \\
S_{i+1,2} &= S_{i,2} \oplus \mathrm{R}(S_{i,1}) \\
S_{i+1,3} &= S_{i,3} \oplus \mathrm{R}(S_{i,2}) \\
S_{i+1,4} &= S_{i,4} \oplus \mathrm{R}(S_{i,3}) \\
S_{i+1,5} &= S_{i,5} \oplus \mathrm{R}(S_{i,4})
\end{aligned}$$

Each round, the ciphertext is output as:

$$C_i = S_{i,1} \oplus S_{i,4} \oplus S_{i,5} \oplus (S_{i,2} \,\&\, S_{i,3}) \oplus m_i$$

2.3 Security Claims

AEGIS-128 and AEGIS-256 claim a security level of respectively 128 and 256 bits for plaintext confidentiality (provided the attacker did not first break the integrity of the scheme, for which a security level of 128 bits is claimed in both cases–cf. [WP14], Section 3). There is no explicit bound on data requirements.

3 Preliminaries

3.1 Linear Biases and Weights

Since we will typically deal with probabilities very close to 1/2, it is convenient to define the *bias* of an event (as a shorcut for the bias of its probability):

Definition 1 (Bias). *The* bias *of a an event E is defined as:*

$$\mathrm{Bias}(E) = 2 \cdot \mathrm{Prob}(E) - 1$$

Definition 2 (Linear Bias). *Consider a function $F : \{0,1\}^n \to \{0,1\}^n$ from n bits to n bits. Given an* input mask $\alpha \in \{0,1\}^n$ *and* output mask $\beta \in \{0,1\}^n$*, the* linear bias *of F with masks α, β, is defined as:*

$$\mathrm{Bias}(\alpha \cdot X \oplus \beta \cdot F(X) = 0)$$

with X uniformly random in $\{0,1\}^n$.

Matsui's classic piling-up lemma [Mat94] is commonly used to combine linear biases together.

Lemma 1 (Piling-up Lemma). *Let $X_1, \ldots, X_n$ be independent random binary variables. Then:*

$$\mathrm{Bias}(X_1 \oplus \cdots \oplus X_n = 0) = \mathrm{Bias}(X_1 = 0) \times \cdots \times \mathrm{Bias}(X_n = 0)$$

In the rest of this article, biases will often be of the form $\pm 2^{-i}$, with i an integer. This leads to the following definitions:

Definition 3 (Weight of an Event). *Let E be an event. The* weight *of E is the positive real:*

$$\mathrm{weight}(E) = -\log_2\big(|\mathrm{Bias}(E)|\big)$$

If the bias is zero, we define the weight as ∞.

Definition 4 (Weight of a Linear Bias). *The* weight *of a linear bias is the weight of its bias. That is, with the previous notations:*

$$\mathrm{weight}(F, \alpha, \beta) = -\log_2\big(|\mathrm{Bias}(\alpha \cdot X \oplus \beta \cdot F(X) = 0)|\big)$$

While the notion of weight is more prevalent in differential than in linear cryptanalysis, we have defined it so that it behaves exactly in the same way: due to Lemma 1, when combining linear characteristics, we simply add their weights together, making computations more readable. Since we will combine linear characteristics repeatedly, the benefits are substantial. Note that finding strong biases means we always want to minimize weights.

3.2 Linear Approximations of Bitwise AND

For x, y two independent uniformly random binary variables, it can be easily checked that their product $x\&y$ can be linearly approximated in four different ways: 0, x, y and $x \oplus y \oplus 1$, each with probability 3/4. In particular, this implies the following lemma, which will be quite useful:

Lemma 2. *Let X, Y be two independent uniformly random variables in $\{0,1\}^n$, and α be a linear mask in $\{0,1\}^n$. Then:*

$$\begin{aligned}
&\text{weight}\big(\alpha \cdot (X\&Y) = 0\big) \\
&= \text{weight}\big(\alpha \cdot (X\&Y \oplus X) = 0\big) \\
&= \text{weight}\big(\alpha \cdot (X\&Y \oplus Y) = 0\big) \\
&= \text{weight}\big(\alpha \cdot (X\&Y \oplus X \oplus Y \oplus 1) = 0\big) \\
&= |\alpha|
\end{aligned}$$

where $|\alpha|$ denotes the Hamming weight of α. The biases are all positive.

4 Linear Biases for AEGIS-128 and AEGIS-256

4.1 Linear Biases between Substates

The output of AEGIS-128 at round i is $C_i = S_{i,1} \oplus S_{i,4} \oplus (S_{i,2} \,\&\, S_{i,3}) \oplus m_i$. Using linear approximations of & in the previous section, this can naturally be approximated as a sum of some substates $S_{i,j}$'s. As a preliminary step towards exhibiting biases in the AEGIS-128 keystream, we point out some useful linear relations between substates $S_{i,j}$'s over three rounds.

Assume that at some round i, three consecutive plaintext blocks m_i, m_{i+1}, m_{i+2} are all-zeros. Denote $S_0 = S_{i,0}, \ldots, S_4 = S_{i,4}$. Then we can compute the value of substate 0 over the three rounds i, $i+1$, $i+2$ as:

$$\begin{aligned}
S_{i,0} &= S_0 \\
S_{i+1,0} &= S_0 \oplus \mathrm{R}(S_4) \\
S_{i+2,0} &= S_0 \oplus \mathrm{R}(S_4) \;\oplus\; \mathrm{R}(S_4 \oplus \mathrm{R}(S_3))
\end{aligned}$$

We are interested in the two differences $S_{i,0} \oplus S_{i+1,0}$ and $S_{i,0} \oplus S_{i+2,0}$. Let us begin with the first:

$$S_0 \oplus S_{i+1,0} = \mathrm{R}(S_4)$$

If we choose any linear mask α, β with $w = \text{weight}(\mathrm{R}, \alpha, \beta)$, then by definition we have:

$$\beta \cdot (S_0 \oplus S_{i+1,0}) = \alpha \cdot S_4 \qquad \text{with weight } w \quad (1)$$

Now consider the second difference:

$$S_{i+2,0} \oplus S_{i,0} = \mathrm{R}(S_4) \;\oplus\; \mathrm{R}(S_4 \oplus \mathrm{R}(S_3))$$

This is the derivative of R at point S_4 with difference $\mathrm{R}(S_3)$. Choose two linear masks β, γ with $w' = \text{weight}(\mathrm{R}, \beta, \gamma)$. By the piling-up lemma, we get:

$$\begin{aligned}
\gamma \cdot (S_{i+2,0} \oplus S_{i,0}) &= \beta \cdot S_4 \;\oplus\; \beta \cdot (S_4 \oplus \mathrm{R}(S_3)) \qquad \text{with weight } 2w' \quad (2) \\
&= \beta \cdot \mathrm{R}(S_3)
\end{aligned}$$

Thus, the contribution of S_4 cancels itself out.

Finally, we can combine the previous linear approximation of R along α, β with (2) to get:

$$\gamma \cdot (S_{i+2,0} \oplus S_{i,0}) = \alpha \cdot S_3 \qquad \text{with weight } w + 2w' \quad (3)$$

Note that the above approximations also hold for $S_{i,1}, \ldots, S_{i,4}$ by shifting all $S_{i,j}$'s involved along j modulo 5. Furthermore the same equalities hold for AEGIS-256 as well, except S_3 and S_4 in all three equations (1), (2), (3) become S_4 and S_5. The main takeaway in all cases is that $S_{i+1,j} \oplus S_{i,j}$ is correlated to $S_{i,j-1}$, while $S_{i+2,j} \oplus S_{i,j}$ is correlated to $S_{i,j-2}$.

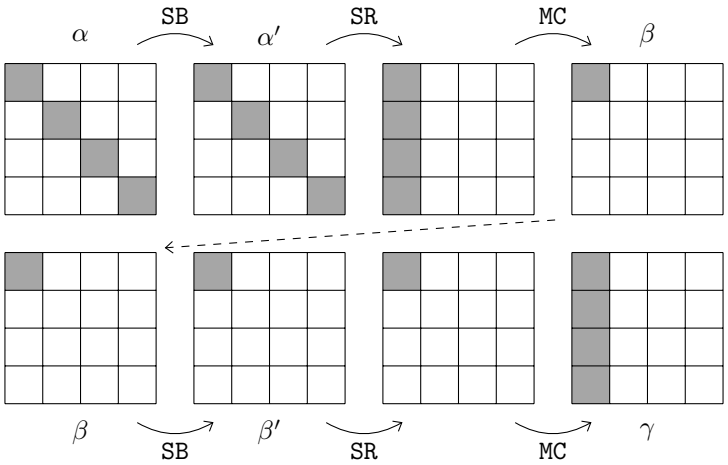

Fig. 2. Linear masks over two rounds of AES. Grey boxes denote active bytes.

In the end, we will want to choose α, γ so as to minimize $w + 2w'$ in (3). This involves considering linear propagation over two rounds of AES. Due to the branching number of 5 of the AES construction [DR01], we will have at least 5 active S-boxes over these two rounds. Moreover, since we want to minimize $w+2w'$, the second round incurs twice the cost, so the optimal configuration would be to have 4 active S-boxes in the first round, and only one in the second round. This is easily achieved: choosing any linear masks at the input and output of a single S-box in the second round, then propagating the masks linearly will have the desired effect (cf. Figure 2). In fact, there are enough degrees of freedom to pass all S-boxes with the optimal linear weight of 3. As a result, we get $w = 4 \cdot 3 = 12$ and $w' = 3$, so $w + 2w' = 18$. Appendix A gives specific values for α, β, γ.

4.2 Biases for AEGIS-128

In this section, we will exhibit a linear bias between the output of AEGIS-128 at rounds i and $i + 2$, assuming that the messages m_i, m_{i+1}, m_{i+2} are all-zeros. Let us define $S_0 = S_{i,0}, \ldots, S_4 = S_{i,4}$.

Choose α, β, γ as in the previous section. Recall that $C_i = S_1 \oplus S_4 \oplus S_2 \,\&\, S_3$. Using §3.2, we can approximate C_i and C_{i+2} as:

$$\gamma \cdot C_i = \gamma \cdot (S_1 \ \oplus \ S_4 \ \oplus \ S_3) \qquad \text{with weight } |\gamma|$$
$$\gamma \cdot C_{i+2} = \gamma \cdot (S_{i+2,1} \ \oplus \ S_{i+2,4} \ \oplus \ S_{i+2,3}) \qquad \text{with weight } |\gamma|$$

It follows from Equation (3) in the previous section that we have:

$$\gamma \cdot (C_i \oplus C_{i+2}) = \alpha \cdot (S_4 \ \oplus \ S_2 \ \oplus \ S_1) \qquad \text{with weight } 3w + 6w' + 2|\gamma|$$

Now, observe that C_i may also be approximated as:

$$\alpha \cdot C_i = \alpha \cdot (S_1 \ \oplus \ S_4 \ \oplus \ S_2) \qquad \text{with weight } |\alpha|$$

We are now approximating C_i bitwise in two different ways. However, as long as α and γ have disjoint support, the two events are independent. In the remainder, we assume this is the case.

If we combine the last two equations together, we get:

$$(\alpha \oplus \gamma) \cdot C_i \oplus \gamma \cdot C_{i+2} = 0 \qquad \text{with weight } 3w + 6w' + |\alpha| + 2|\gamma|$$

This is an absolute bias on the AEGIS-128 keystream. Note that in order to simplify the presentation, we did not keep track of whether the bias is positive or negative; however, this is fixed and known.

The question now becomes how to choose α, γ so as to minimize the weight above. Details of this computation are provided in Appendix A. In the end, we obtain $|\alpha| = 5$, $|\gamma| = 9$, with all S-boxes having optimal linear bias, hence $w = 12$, $w' = 3$ as in the previous section. This yields $3w + 6w' + |\alpha| + 2|\gamma| = 77$.

4.3 Biases for AEGIS-256

Biases on the AEGIS-256 keystream are built essentially in the same way as for AEGIS-128, except the outputs of all three rounds i, $i+1$ and $i+2$ are necessary. Again, we assume $m_i = m_{i+1} = m_{i+2} = 0$. Recall that $C_i = S_1 \oplus S_4 \oplus S_5 \oplus S_2 \,\&\, S_3$.

We use the following approximations:

$$\begin{aligned}
\alpha \cdot C_i &= \alpha \cdot (S_1 \oplus S_4 \oplus S_5) && \text{with weight } |\alpha| \\
\beta \cdot C_i &= \beta \cdot (S_1 \oplus S_4 \oplus S_5 \oplus S_2 \oplus S_3) && \text{with weight } |\beta| \\
\gamma \cdot C_i &= \gamma \cdot (S_1 \oplus S_4 \oplus S_5 \oplus S_2) && \text{with weight } |\gamma| \\
\beta \cdot C_{i+1} &= \beta \cdot (S_{i+1,1} \oplus S_{i+1,4} \oplus S_{i+1,5} \oplus S_{i+1,2} \oplus S_{i+1,3}) && \text{with weight } |\beta| \\
\gamma \cdot C_{i+2} &= \gamma \cdot (S_{i+2,1} \oplus S_{i+2,4} \oplus S_{i+2,5} \oplus S_{i+2,2}) && \text{with weight } |\gamma|
\end{aligned}$$

Using Equation (2) from §4.1, we have:

$$\gamma \cdot (C_i \oplus C_{i+2}) = \beta \cdot (\mathrm{R}(S_5) \oplus \mathrm{R}(S_2) \oplus \mathrm{R}(S_3) \oplus \mathrm{R}(S_0)) \qquad \text{with weight } 8w' + 2|\gamma|$$

On the other hand:

$$\beta \cdot (C_i \oplus C_{i+1}) = \beta \cdot (\mathrm{R}(S_0) \oplus \mathrm{R}(S_3) \oplus \mathrm{R}(S_4) \oplus \mathrm{R}(S_1) \oplus \mathrm{R}(S_2)) \qquad \text{with weight } 2|\beta|$$

Summing the last two equalities yields:

$$\beta \cdot (C_i \oplus C_{i+1}) \oplus \gamma \cdot (C_i \oplus C_{i+2}) = \beta \cdot (\mathrm{R}(S_1) \oplus \mathrm{R}(S_4) \oplus \mathrm{R}(S_5)) \qquad \text{with weight } 8w' + 2|\beta| + 2|\gamma|$$

Now it remains to use Equation (1) to pass through R and get:

$$\beta \cdot (C_i \oplus C_{i+1}) \oplus \gamma \cdot (C_i \oplus C_{i+2}) = \alpha \cdot (S_1 \oplus S_4 \oplus S_5) \qquad \text{with weight } 3w + 8w' + 2|\beta| + 2|\gamma|$$

Finally:

$$\alpha \cdot C_i \oplus \beta \cdot (C_i \oplus C_{i+1}) \oplus \gamma \cdot (C_i \oplus C_{i+2}) = 0 \qquad \text{with weight } 3w + 8w' + |\alpha| + 2|\beta| + 2|\gamma|$$

Now the question is to find α, β, γ minimizing this weight. In fact, we can choose precisely the same α, γ as for AEGIS-128. Indeed, these masks were chosen so as to 1) pass all S-boxes with optimal probability; 2) minimize w as compared to w'; 3) minimize $|\alpha| + 2|\gamma|$ within the previous

constraints. The same criterions are very fitting once again; the only difference is the new β term, but with the previous choices $|\beta| = 3$, so it is nearly optimal as well. As a result, we have a weight of $3 \cdot 12 + 8 \cdot 3 + 5 + 2 \cdot 3 + 2 \cdot 9 = 89$.

Intuition. How the previous linear approximations were chosen so as to cancel each other out, and perhaps more importantly what made such a choice possible, may not be immediately apparent from the description of the linear characteristic itself. As a result, it may be useful to provide some intuition.

We know that $S_j \oplus S_{i+1,j} = \mathrm{R}(S_{j-1})$. From Eq. (2), $S_j \oplus S_{i+2,j} = \mathrm{R}(S_{j-1}) \oplus \mathrm{R}(S_{j-1} \oplus \mathrm{R}(S_{j-2}))$ is linearly correlated to $\mathrm{R}(S_{j-2})$, with the contribution of S_{j-1} cancelling itself out; so we may write $S_j \oplus S_{i+2,j} = \mathrm{D}(\mathrm{R}(S_{j-2}))$, where D is a purely formal notation to indicate an expression that is linearly correlated to the input of D.

On the other hand, if we approximate the & operation in C_i and C_{i+1} linearly along the same mask, and add them together, we can ensure that every $S_{i+1,j}$ is matched with the corresponding S_j, so as a result we can roughly write:

$$C_{i+1} \oplus C_i \approx \quad \mathrm{R}(S_0) \oplus \quad [\mathrm{R}(S_1)] \oplus \quad [\mathrm{R}(S_2)] \oplus \quad \mathrm{R}(S_3) \oplus \quad \mathrm{R}(S_4)$$

where the brackets denote a term that comes from a & operation and thus may be omitted at will by §3.2.

The same reasoning holds for C_{i+2}; and in the end we have:

$$\begin{aligned}
C_i \approx \quad & & S_1 \oplus & [S_2] \oplus & [S_3] \oplus & S_4 \oplus & S_5 \\
C_{i+1} \oplus C_i \approx \quad & \mathrm{R}(S_0) \oplus & [\mathrm{R}(S_1)] \oplus & [\mathrm{R}(S_2)] \oplus & \mathrm{R}(S_3) \oplus & \mathrm{R}(S_4) & \\
C_{i+2} \oplus C_i \approx \quad & [\mathrm{D}(\mathrm{R}(S_0))] \oplus & [\mathrm{D}(\mathrm{R}(S_1))] \oplus & \mathrm{D}(\mathrm{R}(S_2)) \oplus & \mathrm{D}(\mathrm{R}(S_3)) \oplus & & \mathrm{D}(\mathrm{R}(S_5))
\end{aligned}$$

Now take the characteristic $\alpha \xrightarrow{\mathrm{R}} \beta \xrightarrow{\mathrm{R}} \gamma$ from the previous section, which is also a characteristic $\alpha \xrightarrow{\mathrm{R}} \beta \xrightarrow{\mathrm{D}} \gamma$ as can be seen in §4.1, Eq. (2). If the characteristics hold, this tells us that $\alpha \cdot S_k = \beta \cdot \mathrm{R}(S_k) = \gamma \cdot \mathrm{D}(\mathrm{R}(S_k))$. Hence, if we approximate the first line along α, the second along β, and the third along γ, if the linear characteristics hold and we add up everything, any two terms in the same column will cancel each other out. So the question becomes simply how to make an appropriate choice for each bracket in the equations above so that there is 0 or 2 terms in each column. This is exactly what we do in order to construct our linear characteristic, namely:

$$\begin{aligned}
C_i \approx \quad & & S_1 \oplus & & & S_4 \oplus & S_5 \\
C_{i+1} \oplus C_i \approx \quad & \mathrm{R}(S_0) \oplus & \mathrm{R}(S_1) \oplus & \mathrm{R}(S_2) \oplus & \mathrm{R}(S_3) \oplus & \mathrm{R}(S_4) & \\
C_{i+2} \oplus C_i \approx \quad & \mathrm{D}(\mathrm{R}(S_0)) \oplus & & \mathrm{D}(\mathrm{R}(S_2)) \oplus & \mathrm{D}(\mathrm{R}(S_3)) \oplus & & \mathrm{D}(\mathrm{R}(S_5))
\end{aligned}$$

If we look at AEGIS-128 from the same perspective, we can write:

$$\begin{aligned}
C_i \approx \quad & & S_1 \oplus & [S_2] \oplus & [S_3] \oplus & S_4 \\
C_{i+1} \oplus C_i \approx \quad & \mathrm{R}(S_0) \oplus & [\mathrm{R}(S_1)] \oplus & [\mathrm{R}(S_2)] \oplus & \mathrm{R}(S_3) & \\
C_{i+2} \oplus C_i \approx \quad & [\mathrm{D}(\mathrm{R}(S_0))] \oplus & [\mathrm{D}(\mathrm{R}(S_1))] \oplus & \mathrm{D}(\mathrm{R}(S_2)) \oplus & & \mathrm{D}(\mathrm{R}(S_4))
\end{aligned}$$

After removing the second line entirely, the approximation we made is:

$$\begin{aligned}
C_i \approx \quad & & S_1 \oplus & S_2 \oplus & S_4 \\
C_{i+2} \oplus C_i \approx \quad & & \mathrm{D}(\mathrm{R}(S_1)) \oplus & \mathrm{D}(\mathrm{R}(S_2)) \oplus & \mathrm{D}(\mathrm{R}(S_4))
\end{aligned}$$

4.4 Exploiting the Keystream Biases

In the previous two sections, we have assumed that at some round i, three consecutive plaintexts m_i, m_{i+1}, m_{i+2} are all-zeros. From there, we have shown the existence of an absolute bias of the form:

$$\mathrm{Bias}(\alpha \cdot C_i \oplus \beta \cdot C_{i+1} \oplus \gamma \cdot C_{i+2} \oplus b = 0) = 2^{-w}$$

In other words, we have built a distinguisher on the AEGIS keystream. However, if we no longer assume $m_{i+2} = 0$, then at round $i+2$, the only difference in the output of the cipher is that m_{i+2} is `XOR`-ed into the ciphertext C_{i+2}. As a result, we have:

$$\text{Bias}(\alpha \cdot C_i \oplus \beta \cdot C_{i+1} \oplus \gamma \cdot C_{i+2} \oplus \gamma \cdot m_{i+2} \oplus b = 0) = 2^{-w}$$

Thus, the observable value $\alpha \cdot C_i \oplus \beta \cdot C_{i+1} \oplus \gamma \cdot C_{i+2}$ directly leaks information about $\gamma \cdot m_{i+2}$.

This leads to the following attack scenario. Assume the same three consecutive plaintext blocks 0, 0, m are encrypted 2^{2w} times in total, independently of the keys and nonces used. Then an attacker having access to that data would deduce the value of $\gamma \cdot m$ with good probability, just by counting the occurences of the event $\alpha \cdot C_i \oplus \beta \cdot C_{i+1} \oplus \gamma \cdot C_{i+2} = 0$ on the fly. However, the data requirements make this attack impractical, since 2^{154} and 2^{188} encryptions would be required respectively for AEGIS-128 and AEGIS-256 in order to exploit a single bias (as opposed to a multilinear approach). In Appendix B, we try to capture linear propagation in AEGIS-128 more accurately, in order to evaluate to what extent data requirements could be lowered; we conclude that AEGIS-128 seems to resist straightforward improvements of our attack, as 2^{140} data is still required.

5 Conclusion

In this article, we have constructed linear biases in the keystream of AEGIS-128 and AEGIS-256. These biases stem from dependencies between surprisingly few consecutive rounds: for AEGIS-128, linear biases exist between the outputs of rounds i and $i+2$; while for AEGIS-256, three consecutive rounds are enough. Our main result is the construction of a linear mask with bias 2^{-89} on the keystream of AEGIS-256. This bias can be exploited to recover bits of information on a partially known plaintext encrypted 2^{188} times, regardless of the keys involved. While the biases remain too low to be a threat in practice, they are vastly superior to any generic attack, and point out an unexpected property in the keystream of AEGIS.

Acknowledgments

The author would like to thank all members of the ANSSI cryptography laboratory, especially Thomas Fuhr and Henri Gilbert, for their valuable comments and insights on this article.

References

ABP+13. N. AlFardan, D. J. Bernstein, K. G. Paterson, B. Poettering, and J. Schuldt. On the security of RC4 in TLS. In USENIX Security Symposium (2013), Presented at FSE 2013 as an invited talk by Daniel J. Bernstein, available at http://www.isg.rhul.ac.uk/tls/, 2013.

BDPA12. Guido Bertoni, Joan Daemen, Michal Peeters, and Gilles Assche. Duplexing the sponge: Single-pass authenticated encryption and other applications. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography*, volume 7118 of *Lecture Notes in Computer Science*, pages 320–337. Springer Berlin Heidelberg, 2012.

cae13. CAESAR– Competition for Authenticated Encryption: Security, Applicability, and Robustness. General secretary Daniel J. Bernstein, information available at http://competitions.cr.yp.to/caesar.html, 2013.

CHJ02. Don Coppersmith, Shai Halevi, and Charanjit Jutla. Cryptanalysis of stream ciphers with linear masking. In Moti Yung, editor, *Advances in Cryptology CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 515–532. Springer Berlin Heidelberg, 2002.

DR99. Joan Daemen and Vincent Rijmen. AES proposal: Rijndael. Advanced Encryption Standard submission, available at http://jda.noekeon.org/, 1999.

DR01. Joan Daemen and Vincent Rijmen. The wide trail design strategy. In Bahram Honary, editor, *Cryptography and Coding*, volume 2260 of *Lecture Notes in Computer Science*, pages 222–238. Springer Berlin Heidelberg, 2001.

Mat94. Mitsuru Matsui. Linear cryptanalysis method for des cipher. In Tor Helleseth, editor, *Advances in Cryptology EUROCRYPT 93*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer Berlin Heidelberg, 1994.

Nik14. Ivica Nikolić. Tiaoxin–346. CAESAR submission, http://competitions.cr.yp.to/round1/tiaoxinv1.pdf, 2014.

WP14. Hongjun Wu and Bart Preneel. AEGIS: A fast authenticated encryption algorithm. CAESAR submission, updated from Cryptology ePrint Archive Report 2013/695, updated from SAC 2013 version, http://competitions.cr.yp.to/round1/aegisv1.pdf, 2014.

YWH+14. Dingfeng Ye, Peng Wang, Lei Hu, Liping Wang, Yonghong Xie, Siwei Sun, and Ping Wang. PAES v1: Parallelizable authenticated encryption schemes based on AES round function. CAESAR submission, http://competitions.cr.yp.to/round1/paesv1.pdf, 2014.

Appendix A: Values of α, β, γ

Consider the situation depicted on Figure 3, where the linear characteristic $\alpha \to \beta \to \gamma$ spans two rounds of AES (without key addition). We are trying to minimize $|\alpha| + 2|\gamma|$, while passing all S-boxes with optimal linear probability.

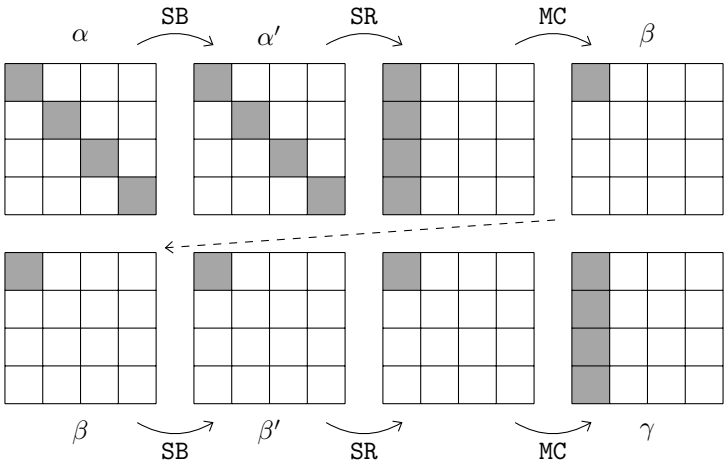

Fig. 3. Linear masks over two rounds of AES. Grey boxes denote active bytes.

First, we look for β' minimizing $|\gamma|$. With little-endian hexadecimal notations, it turns out only one value $\beta' =$ `0e` reaches the minimum with $|\gamma| = 9$ (γ equals `f, 8, c, 5` along one column). For this value, five choices of β allow us to pass the S-box with weight 3: $\beta =$ `09`, `31`, `38`, `c8`, or `f9`. For each of these values, we compute α', then look for the minimal size of α such that all four S-boxes are passed with optimal probability. We find $|\alpha| = 5$, for $\beta =$ `38` (α equals `4, 3, 2, 2` along the diagonal). Observe that $|\alpha|$ is at least 4 since it has to span 4 S-boxes. Since we followed the only way to have $|\gamma| = 9$, which is optimal; $|\alpha|$ is within 1 of being optimal; and we are trying to minimize $|\alpha| + 2|\gamma|$, we have found the unique optimal choice.

More accurately, it is the unique optimal choice once we have fixed our choice for the active S-box in the second round. In fact, we could choose any one of the other 15 S-boxes, and use the exact same values of linear masks β, β' at the entrance of that S-box (namely, `38` and `0e`). Indeed, the circulant nature of the AES `MixColums` matrix means that a given mask at the input (resp. output) of one S-box will propagate to a permutation of the same masks at the output (resp. input) of the previous (resp. next) layer of S-boxes, and hence in our case will yield the same sizes for α and γ. Thus there are 16 choices for α, γ with identical properties for our purpose; one per choice of active S-box in the middle round.

Appendix B: Refined Linear Model of AEGIS-128

In §4.2, we have found a linear mask with bias 2^{-77} on the AEGIS-128 keystream. Detecting this bias would require 2^{154} data, which is significantly more than is sensible with a security parameter of 128 bits. However, the bias on an actual AEGIS-128 keystream may be slightly different, due to the independence assumptions required by our analysis. Additionaly, there may be linear hull effects strengthening or weakening the bias. In any case, other biases of comparable strength undoubtedly exist, and could be exploited in a multilinear attack. With all this in mind, it may be worth wondering whether the 2^{154} data requirement has some chance of being brought down below 2^{128}.

It so happens that for AEGIS-128, there is a fairly elegant way of simultaneously taking into account many of the effects listed above. In our previous analysis, we used standard linear cryptanalysis techniques to follow the propagation of a bias along a few AES-based transformations. This amounts to modelling the transformations in a certain way, materialized by independence assumptions. However in the case of AEGIS-128, large parts of the transformations can be computed with complete accuracy by looking at byte distributions, without the need to model anything.

If we recap the previous analysis in §4.2, we approximate C_i and $C_i \oplus C_{i+2}$ linearly, and from there we obtain the following two sums:

$$\begin{aligned} & S_1 \ \oplus \ S_2 \ \oplus \ S_4 \\ \textit{and}: \quad & \mathrm{R}(S_2) \oplus \mathrm{R}(S_2 \oplus \mathrm{R}(S_1)) \oplus \mathrm{R}(S_3) \oplus \mathrm{R}(S_3 \oplus \mathrm{R}(S_2)) \oplus \mathrm{R}(S_0) \oplus \mathrm{R}(S_0 \oplus \mathrm{R}(S_4)) \\ = \ & \mathrm{D}(\mathrm{R}(S_1)) \ \oplus \ \mathrm{D}(\mathrm{R}(S_2)) \ \oplus \ \mathrm{D}(\mathrm{R}(S_4)) \end{aligned}$$

where D is a purely formal notation denoting the fact that its input and output are linearly correlated (cf. §4.1); then we use the fact that X and $\mathrm{D}(\mathrm{R}(X))$ are correlated. More precisely, we first relate X to $\mathrm{R}(X)$, then $\mathrm{R}(X)$ to $\mathrm{D}(\mathrm{R}(X))$. Thus the propagation is decomposed in two steps, which we can picture as:

$$S_1 \oplus S_2 \oplus S_4 \quad \rightarrow \quad \mathrm{R}(S_1) \oplus \mathrm{R}(S_2) \oplus \mathrm{R}(S_4) \quad \rightarrow \quad \mathrm{D}(\mathrm{R}(S_1)) \oplus \mathrm{D}(\mathrm{R}(S_2)) \oplus \mathrm{D}(\mathrm{R}(S_4))$$

So our propagation "factors" through the value $\mathrm{R}(S_1) \oplus \mathrm{R}(S_2) \oplus \mathrm{R}(S_4)$: that is to say, all information we had on $S_1 \oplus S_2 \oplus S_4$ is first translated as information on $\mathrm{R}(S_1) \oplus \mathrm{R}(S_2) \oplus \mathrm{R}(S_4)$; after which only information on $\mathrm{R}(S_1) \oplus \mathrm{R}(S_2) \oplus \mathrm{R}(S_4)$ is used to deduce information on $\mathrm{D}(\mathrm{R}(S_1)) \oplus \mathrm{D}(\mathrm{R}(S_2)) \oplus \mathrm{D}(\mathrm{R}(S_4))$. Moreover, with our linear masks, only a single S-box is active in $\mathrm{R}(S_1) \oplus \mathrm{R}(S_2) \oplus \mathrm{R}(S_4)$, so actually the whole propagation factors through the value of $\mathrm{R}(S_1) \oplus \mathrm{R}(S_2) \oplus \mathrm{R}(S_4)$ on a single byte.

The idea for our new model is that we are going to compute the actual distribution of $\mathrm{R}(S_1) \oplus \mathrm{R}(S_2) \oplus \mathrm{R}(S_4)$ on one byte from the knowledge of $C_i = S_1 \oplus S_2 \,\&\, S_3 \oplus S_4$. Then we are going to use this full distribution, rather than a single linear mask, to compute a distribution of $S_{i+2,3} \oplus S_3 \ \oplus \ S_{i+2,4} \oplus S_4 \ \oplus \ S_{i+2,1} \oplus S_1$, which is our linear approximation of $C_i \oplus C_{i+2}$ along a linear mask γ. Thus we hope to compute the bias of $\gamma \cdot (C_i \oplus C_{i+2})$ more accurately. A good motivation for this model is that the two steps above: linking knowledge of C_i to the distribution of $\mathrm{R}(S_1) \oplus \mathrm{R}(S_2) \oplus \mathrm{R}(S_4)$ on one byte; and then the distribution of $\mathrm{R}(S_1) \oplus \mathrm{R}(S_2) \oplus \mathrm{R}(S_4)$ on one byte to the bias of the linear approximation of $\gamma \cdot (C_i \oplus C_{i+2})$, can be computed with perfect precision within complexity at most 2^{32}, as we show below. Hence the only loss of precision results from "factoring" through $\mathrm{R}(S_1) \oplus \mathrm{R}(S_2) \oplus \mathrm{R}(S_4)$; but as we saw in the previous paragraph, we were already making this approximation when we used standard linear characteristic techniques.

Thus, assume we know some specific value for $C_i = S_1 \oplus S_2 \,\&\, S_3 \oplus S_4$. Then we can actually compute the distribution of a single byte of $\mathrm{R}(S_1) \oplus \mathrm{R}(S_2) \oplus \mathrm{R}(S_4)$ with full precision. Indeed, if we denote by `SB` the `SubBytes` layer of AES, from $S_1 \oplus S_2 \,\&\, S_3 \oplus S_4$ we can compute the distribution of $\mathtt{SB}(S_1) \oplus \mathtt{SB}(S_2) \oplus \mathtt{SB}(S_4)$ in an exact manner, since each byte depends only on the value S_1, S_2, S_3, S_4 on the same byte; so we need only guess 4 bytes simultaneously.

From there, we can also compute the distribution of $\mathrm{R}(S_1) \oplus \mathrm{R}(S_2) \oplus \mathrm{R}(S_4)$ on a single byte exactly, since it is simply the independent sum of the previous byte distributions through the `MixColumns` matrix. Moreover, in the end, this distribution depends on only 16 bytes in total, which is 128 bits, so we can simply count how many choices lead to a specific value using a 128-bit integer, and the resulting distribution is prefectly precise. Thus, from knowledge of $C_i = S_1 \oplus S_2 \,\&\, S_3 \oplus S_4$, it is possible to compute the distribution of $\mathrm{R}(S_1) \oplus \mathrm{R}(S_2) \oplus \mathrm{R}(S_4)$ on one byte with full precision.

Now for the second step of the propagation, we want to compute the distribution of the following value (i.e. the linear approximation of $C_i \oplus C_{i+2}$ along the mask γ):

$$\begin{aligned} & S_{i+2,3} \oplus S_3 \ \oplus \ S_{i+2,4} \oplus S_4 \ \oplus \ S_{i+2,1} \oplus S_1 \\ = \ & \mathrm{R}(S_2) \oplus \mathrm{R}(S_2 \oplus \mathrm{R}(S_1)) \ \oplus \ \mathrm{R}(S_3) \oplus \mathrm{R}(S_3 \oplus \mathrm{R}(S_2)) \ \oplus \ \mathrm{R}(S_0) \oplus (S_0 \oplus \mathrm{R}(S_4)) \end{aligned}$$

from the known distribution of:

$$\mathrm{R}(S_1) \oplus \mathrm{R}(S_2) \oplus \mathrm{R}(S_4)$$

More precisely, we are interested in the distribution of the first value before `MixColumns` (which is the last operation applied to each component), on a single byte, so R reduces to one S-box layer (and a permutation of the bytes).

At first sight, it suffices to guess the values of all $\mathrm{R}(S_i)$'s on this one byte. This only involves guessing 5 bytes, requiring 2^{40} operations, which is reasonable. However a better algorithm is possible by observing that the contribution of S_0 and S_4 is independent from the rest on both sides. As a result, it suffices to compute these two distributions separately, then add them together:

$$\begin{aligned} \mathrm{R}(S_1) \oplus \mathrm{R}(S_2) \to \mathrm{R}(S_2) \oplus \mathrm{R}(S_2 \oplus \mathrm{R}(S_1)) \ \oplus \ \mathrm{R}(S_3) \oplus \mathrm{R}(S_3 \oplus \mathrm{R}(S_2)) \\ \mathrm{R}(S_4) \to \mathrm{R}(S_0) \oplus (S_0 \oplus \mathrm{R}(S_4)) \end{aligned}$$

Thus the complexity drops down to 2^{32} operations.

The end result is that for a fixed value of $C_i = S_1 \oplus S_2 \,\&\, S_3 \oplus S_4$, we can compute the distribution of $\mathrm{R}(S_1) \oplus \mathrm{R}(S_2) \oplus \mathrm{R}(S_4)$ on one byte without any approximation. Then from this distribution, we can compute the distribution of one byte of the linear approximation of $C_i \oplus C_{i+2}$ before `MixColumns`, which is what we measure from $C_i \oplus C_{i+2}$ using our linear masks, modulo the cost of the linear approximation along γ, which is $2|\gamma|$.

We implemented this model, and results correlate fairly well with our previous anaysis. In particular, we recover the fact that the values of β' we chose yields the strongest bias, although one other value seems as strong (namely `12`), which is not too surprising since it is one of the two second-best candidates as far as minimizing $|\gamma|$. The main difference is that we find a bias close to 2^{-72} when we fix C_i to some random value and measure $\gamma \cdot (C_i \oplus C_{i+2})$ according to our model, rather than 2^{-77} when we used pure linear masking. We surmise this is mostly due to more information being taken into account at the input, resulting overall in more information at the output; although both steps of the new model behave slightly better than expected.

On the other hand, few output masks yield biases in this vicinity. If we exploit the best bias at 2^{-72}, across all 16 possible second-round S-boxes (cf. Appendix A), $2^{144-4} = 2^{140}$ data would still be required to mount an attack. With additional improvements, one could hope to further reduce data requirements, but at this point it seems very unlikely that data requirements could fall below 2^{128}. Thus, the main conclusion of our model seems to be that AEGIS-128 remains resistant to straightforward improvements of our attack.