Weak Keys for AEZ, and the External Key Padding Attack

Bart Mennink[1,2]

[1] Dept. Electrical Engineering, ESAT/COSIC, KU Leuven, and iMinds, Belgium
bart.mennink@esat.kuleuven.be
[2] Digital Security Group, Radboud University, Nijmegen, The Netherlands
b.mennink@cs.ru.nl

Abstract. AEZ is one of the third round candidates in the CAESAR competition. We observe that the tweakable blockcipher used in AEZ suffers from structural design issues in case one of the three 128-bit subkeys is zero. Calling these keys "weak," we show that a distinguishing attack on AEZ with weak key can be performed in at most five queries. Although the fraction of weak keys, around 3 out of every 2^{128}, seems to be too small to violate the security claims of AEZ in general, they do reveal unexpected behavior of the scheme in certain use cases. We derive a potential scenario, the "external key padding," where a user of the authenticated encryption scheme pads the key externally before it is fed to the scheme. While for most authenticated encryption schemes this would affect the security only marginally, AEZ turns out to be completely insecure in this scenario due to its weak keys. These observations open a discussion on the significance of the "robustness" stamp, and on what it encompasses.

Keywords. AEZ, tweakable blockcipher, weak keys, attack, external key padding, robustness.

1 Introduction

Authenticated encryption aims to offer both privacy and authenticity of data. The ongoing CAESAR competition [8] targets the development of a portfolio of new, solid, authenticated encryption schemes. It received 57 submissions, 30 candidates advanced to the second round, and recently, 16 of those advanced to the third round.

AEZ is an authenticated encryption scheme by Hoang, Krovetz, and Rogaway [18]. In this work we focus on AEZ v4, the latest version that has been submitted to CAESAR [17]. The addendum "v4" will be omitted for brevity. We remark that our findings can also be generalized to versions v2 and v3, despite the major revisions that have been made in the key scheduling. Our attacks do not apply to v1, because it differs from v4 not only in the key scheduling but also in the encryption mode itself.

AEZ is designed as a "robust authenticated encryption (RAE) scheme" [18]; this informally means that it achieves privacy and authenticity as good as possible even in the case of nonce-reuse. It moreover implies that it is secure in case of release of unverified plaintext [2]. The designers of AEZ claim that it is a RAE scheme as long as the query complexity does not exceed 2^{55} and the time complexity does not exceed 2^{128} [17].

On the other hand, robustness implies nothing for more "alternative" attacks, such as key recovery attacks, related-key attacks, and others. In [13], Fuhr et al. derived a key recovery attack on AEZ v3 in complexity $2^{n/2}$; not breaking the claimed security, but definitely an unexpected security property. In response to the observation by Fuhr et al., the designers of AEZ performed a major revision from AEZ v3 to AEZ v4 in order to mitigate the attack. Chaigneau and Gilbert [9], however, demonstrated that v4.1 is still vulnerable to a key recovery attack with a similar complexity to that of [13].

Beyond [9, 13], no analysis on AEZ has appeared. In this work, we will investigate the underlying tweakable blockcipher of AEZ and notice that it shows remarkable behavior for certain structured sets of keys. We will show how these weak keys can be used to attack the AEZ mode and to distinguish it from a random primitive in constant time. We will additionally discuss a specific use case of AEZ where its weak keys can be exploited.

1.1 Weak Keys

AEZ allows for arbitrarily-sized keys, and transforms them into three subkeys of 128 bits using a key derivation function:

$$I\|J\|L \longleftarrow \begin{cases} K \text{ if } |K| = 384\,, \\ \text{BLAKE2b}(K) \text{ otherwise}\,. \end{cases}$$

In other words, if the key is already 384 bits long, it is simply padded into $I\|J\|L$; otherwise, it is first hashed via BLAKE2b [5]. This is done deliberately, as the authors state [17]: "We dispense with calling BLAKE2b if the key K is already $3 \cdot 128$ bits."

We will show that if one of the three subkeys I, J, L equals 0^{128}—call a key K for which this is the case "weak"—AEZ can be distinguished from random in at most two evaluations if it is known which subkey equals 0^{128} and at most five evaluations otherwise. The attack relies on the fact that for weak keys the tweakable blockcipher used in AEZ is completely insecure. In more detail, by explicitly writing out this tweakable blockcipher, as we have done in Section 3.1,[3] one finds that if I, J, or L equals 0^{128}, one can identify multiple tweaks for which the tweakable blockcipher collides.

A simple computation shows that, if we consider keys of length 384 bits, $3 \cdot 2^{256} - 3 \cdot 2^{128} + 1 \approx 3 \cdot 2^{256}$ of those are weak. Regarding keys of size different

[3] This explicit description may contribute to a better understanding of the primitive used in AEZ, and may be of independent interest.

from 384, assuming that BLAKE2b is a random oracle (see [12,15] for the latest analysis of BLAKE2b) approximately 3 out of 2^{128} keys result in a subkey 0^{128}. Although this in itself does not break the security claims of AEZ, the observation testifies of a more structural weakness in AEZ, namely that *the underlying tweakable blockcipher is not secure (for these weak keys)*.

1.2 External Key Padding

Focusing on keys of length different from 384 bits, a key is weak if $I\|J\|L =$ BLAKE2b(K) satisfies that I, J, or L equals 0^{128}. This set of weak keys is rather unstructured; hitting a weak key is a mere coincidence. As a matter of fact, calling these keys "weak" is debatable in the first place.

For keys of length exactly 384 bits, the situation is completely different. We will illustrate this via a potential use case, which we call the "external key padding." At a high level, this scenario covers the case where the user of AEZ pads the key himself prior to feeding it to the scheme. Partly attributed to the key scheduling of AEZ, this would result in an omission of the evaluation of BLAKE2b. Above-mentioned weak key attacks can then be used to distinguish AEZ from random in case of external key padding. Remarkably, for "ordinary" authenticated encryption schemes (such as [3,7,19,21,22,26]), external key padding would only have a marginal influence, mostly because the scheme already pads the key itself in the first place.

A simple patch for this use case would be to *always* hash the key through BLAKE2b, regardless of the size of K. Unfortunately, this patch does not resolve the structural design issues the tweakable blockcipher of AEZ suffers from, and other problematic use cases may exist.

1.3 Outline

A high-level description of the AEZ mode is given in Section 2. We discuss the AEZ tweakable blockcipher primitive, as well as its weak key issues, in Section 3. The weak key attacks on AEZ are discussed in Section 4. We discuss the external key padding scenario and the corresponding attack in Section 5. The work is concluded in Section 6.

2 AEZ

We will describe the interface and security model of AEZ in Section 2.1, and give a high-level description of AEZ in Section 2.2.

2.1 Interface and Security Model

AEZ [17,18] is an authenticated encryption scheme that consists of an encryption function $\mathcal{E}$ and a decryption function $\mathcal{D}$. The encryption $\mathcal{E}$ gets as input a key, nonce, associated data, tag size, and message, and outputs an expanded ciphertext. The decryption $\mathcal{D}$ operates the opposite way; it gets as input a key, nonce, associated data, tag size, and expanded ciphertext, and it outputs either a message or a dedicated $\perp$ symbol. More formally, for some finite key space $\mathcal{K} \subset \{0,1\}^*$,

$$\begin{aligned}
&\mathcal{E} : \mathcal{K} \times \{0,1\}^* \times \{0,1\}^* \times \mathbb{N} \times \{0,1\}^* \to \{0,1\}^* \,, \\
&\quad (K, N, A, \tau, M) \mapsto C \in \{0,1\}^{|M|+\tau} \,, \\
&\mathcal{D} : \mathcal{K} \times \{0,1\}^* \times \{0,1\}^* \times \mathbb{N} \times \{0,1\}^* \to \{0,1\}^* \cup \{\perp\} \,, \\
&\quad (K, N, A, \tau, C) \mapsto M/\perp \,,
\end{aligned}$$

where $\mathcal{D}$ is required to satisfy that

$$\mathcal{D}(K, N, A, \tau, \mathcal{E}(K, N, A, \tau, M)) = M$$

for any K, N, A, τ, M.

AEZ is introduced alongside the security model called "robust authenticated encryption (RAE)," and we will describe it in own terminology. Throughout, $x \xleftarrow{\$} \mathcal{X}$ means that x gets sampled uniformly at random from a finite set $\mathcal{X}$. An adversary $\mathcal{A}$ is a probabilistic algorithm that has access to one or more oracles $\mathcal{O}$, denoted $\mathcal{A}^{\mathcal{O}}$. By $\mathcal{A}^{\mathcal{O}} = 1$ we denote the event that $\mathcal{A}$, after interacting with $\mathcal{O}$, outputs 1.

Let $K \xleftarrow{\$} \mathcal{K}$ be a uniformly randomly drawn key. Denote by π a random injection function with the same interface as $\mathcal{E}_K$. More detailed, π is a family of random functions indexed by $(N, A, \tau) \in \{0,1\}^* \times \{0,1\}^* \times \mathbb{N}$, and a query $\pi(N, A, \tau, M)$ is responded with a $C \in \{0,1\}^{|M|+\tau}$. A decryption query $\pi^{-1}(N, A, \tau, C)$ is responded with either the unique M such that $\pi(N, A, \tau, M) = C$, or with $\perp$ if no such M exists. We refer to [18] for the details.

We define the RAE security of AEZ as

$$\mathbf{Adv}^{\mathrm{rae}}_{\mathrm{AEZ}}(\mathcal{A}) = \left| \mathbf{Pr}_K \left(\mathcal{A}^{\mathcal{E}_K, \mathcal{D}_K} = 1 \right) - \mathbf{Pr}_\pi \left(\mathcal{A}^{\pi, \pi^{-1}} = 1 \right) \right| \,,$$

where the probabilities are taken over the randomness of K, π, and the random choices of $\mathcal{A}$. The resources of $\mathcal{A}$ are usually bounded in terms of (q, ℓ, t), where q is the maximum queries to the construction oracle, each query is of length at most ℓ, and $\mathcal{A}$ runs in time t.

2.2 High-Level Description of AEZ

AEZ takes as input an arbitrarily sized key $K \in \{0,1\}^*$, and performs all of its procedures with three keys $I, J, L \in \{0,1\}^{128}$, where

$$I \| J \| L \longleftarrow \begin{cases} K \text{ if } |K| = 384 \,, \\ \text{BLAKE2b}(K) \text{ otherwise} \,. \end{cases} \tag{1}$$

AEZ then evaluates an algorithm depending on the size of M:[4]

[4] The interfaces of the underlying algorithms have been slightly modified for the sake of simplicity.

- If $|M| = 0$, it evaluates AEZ-prf$(I\|J\|L, N, A, \tau)$;
- If $|M| > 0$:
  - If $|M| < 256 - \tau$, it evaluates Encipher-AEZ-tiny$(I\|J\|L, N, A, \tau, M)$;
  - If $|M| \geq 256 - \tau$, it evaluates Encipher-AEZ-core$(I\|J\|L, N, A, \tau, M)$.

Each of these algorithms starts with an evaluation of the AEZ-hash algorithm, a multi-layer PMAC-style MAC function that transforms (τ, N, A) into a 128-bit mask $\mathit{\Delta}$. In this work, we are specifically interested in AEZ-hash and Encipher-AEZ-core. In more detail, in Section 4, we will describe three weak key attacks on Encipher-AEZ-core: two of which directly concern the Encipher-AEZ-core algorithm, one of which operates via AEZ-hash. The latter attack can be performed equivalently well via AEZ-prf and Encipher-AEZ-tiny, as the three algorithms rely on AEZ-hash in an identical way.

The four sub-algorithms of AEZ internally use a tweakable blockcipher

$$\widetilde{E} : \{0,1\}^{3\cdot 128} \times \mathcal{T} \times \{0,1\}^{128} \to \{0,1\}^{128}\,, \tag{2}$$

that gets as input a key $I\|J\|L \in \{0,1\}^{3\cdot 128}$, a tweak $(j,i) \in \mathcal{T} := \big(\{-1,0\} \times [0..7]\big) \cup \big(\mathbb{N}^{+} \times \mathbb{N}\big)$, and bijectively transforms a plaintext X into a ciphertext $\widetilde{E}^{j,i}_{I\|J\|L}(X)$. We will elaborate on the tweakable blockcipher of AEZ in Section 3.

AEZ-hash. We will use AEZ-hash for the simplified case where $|N| = |A| = 128$; AEZ-hash for this case is given in Algorithm 1. Our attack generalizes to arbitrarily-sized nonces and associated data.

Algorithm 1 AEZ-hash

```
Input: $(I\|J\|L, \tau, N, A)$ with $|N| = |A| = 128$
Output: $\mathit{\Delta} \in \{0,1\}^{128}$
$\mathit{\Delta}_1 \leftarrow \widetilde{E}^{3,1}_{I\|J\|L}(\langle\tau\rangle_{128})$        ▷ $\langle\tau\rangle_{128}$ is the encoding of $\tau$ as an 128-bit string
$\mathit{\Delta}_2 \leftarrow \widetilde{E}^{4,1}_{I\|J\|L}(N)$
$\mathit{\Delta}_3 \leftarrow \widetilde{E}^{5,1}_{I\|J\|L}(A)$
return $\mathit{\Delta} = \mathit{\Delta}_1 \oplus \mathit{\Delta}_2 \oplus \mathit{\Delta}_3$
```

Encipher-AEZ-core. We will describe our attacks for messages M such that $384 \leq |M|+\tau < 511$, and Encipher-AEZ-core for this case is given in Algorithm 2

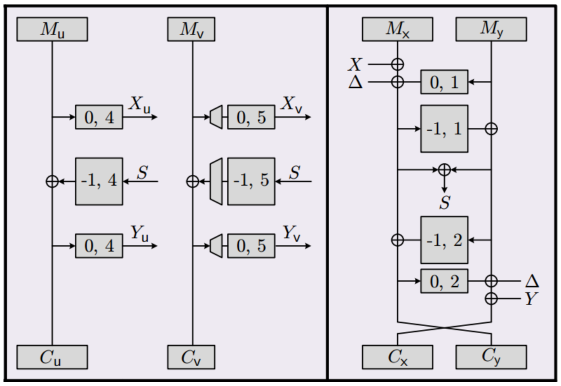


Fig. 1: AEZ for messages such that $384 \leq |M|+\tau < 511$ [17]. Here, the message is padded as $M_{\mathsf{u}} \| M_{\mathsf{v}} \| M_{\mathsf{x}} \| M_{\mathsf{y}} \leftarrow M \| 0^{\tau}$, where $|M_{\mathsf{u}}| = |M_{\mathsf{x}}| = |M_{\mathsf{y}}| = 128$ and $0 \leq |M_{\mathsf{v}}| < 127$. The mask Δ is computed as $\Delta \leftarrow$ AEZ-hash$(I\|J\|L, \tau, N, A)$. A box with inscription j, i represents an evaluation of $\widetilde{E}^{j,i}_{I\|J\|L}$. A trapezoid represents either chopping or 10^*-padding, depending on the direction.

and Figure 1. We remark that the attacks can easily be generalized to any M such that $|M| \geq 256 - \tau$.

Algorithm 2 Encipher-AEZ-core

```
Input: $(I\|J\|L, N, A, \tau, M)$ with $384 \leq |M| + \tau < 511$
Output: $C \in \{0,1\}^{|M|+\tau}$
$\Delta \leftarrow$ AEZ-hash$(I\|J\|L, \tau, N, A)$                ▷ See Algorithm 1
$M_{\mathsf{u}}\|M_{\mathsf{v}}\|M_{\mathsf{x}}\|M_{\mathsf{y}} \leftarrow M\|0^{\tau}$, where $|M_{\mathsf{u}}| = |M_{\mathsf{x}}| = |M_{\mathsf{y}}| = 128$ and $0 \leq |M_{\mathsf{v}}| < 127$
$X \leftarrow \widetilde{E}^{0,4}_{I\|J\|L}(M_{\mathsf{u}}) \oplus \widetilde{E}^{0,5}_{I\|J\|L}(M_{\mathsf{v}}10^*)$
$S_{\mathsf{x}} \leftarrow M_{\mathsf{x}} \oplus \Delta \oplus X \oplus \widetilde{E}^{0,1}_{I\|J\|L}(M_{\mathsf{y}})$ ; $S_{\mathsf{y}} \leftarrow M_{\mathsf{y}} \oplus \widetilde{E}^{-1,1}_{I\|J\|L}(S_{\mathsf{x}})$
$S \leftarrow S_{\mathsf{x}} \oplus S_{\mathsf{y}}$
$C_{\mathsf{u}} \leftarrow M_{\mathsf{u}} \oplus \widetilde{E}^{-1,4}_{I\|J\|L}(S)$ ; $C_{\mathsf{v}} \leftarrow M_{\mathsf{v}} \oplus \widetilde{E}^{-1,5}_{I\|J\|L}(S)$
$Y \leftarrow \widetilde{E}^{0,4}_{I\|J\|L}(C_{\mathsf{u}}) \oplus \widetilde{E}^{0,5}_{I\|J\|L}(C_{\mathsf{v}}10^*)$
$C_{\mathsf{y}} \leftarrow S_{\mathsf{x}} \oplus \widetilde{E}^{-1,2}_{I\|J\|L}(S_{\mathsf{y}})$ ; $C_{\mathsf{x}} \leftarrow S_{\mathsf{y}} \oplus \Delta \oplus Y \oplus \widetilde{E}^{0,2}_{I\|J\|L}(C_{\mathsf{y}})$
return $C_{\mathsf{u}}\|C_{\mathsf{v}}\|C_{\mathsf{x}}\|C_{\mathsf{y}}$
```

3 AEZ Tweakable Blockcipher

We will elaborate on the tweakable blockcipher used in AEZ in Section 3.1, and describe structured sets of weak keys for it in Section 3.2.

3.1 Design

The tweakable blockcipher used in AEZ is internally constructed from the AES round function [10]. Define the *keyless* AES round function $\mathrm{aesr}(X)$ as

$$\mathrm{aesr}(X) = \mathrm{MixColumns} \circ \mathrm{ShiftRows} \circ \mathrm{SubBytes}(X)\,.$$

AEZ uses the two blockciphers AES4 and AES10, where for $r \in \{4, 10\}$,

$$\mathrm{AES}r_{K_0,K_1,\ldots,K_r}(X) = \mathrm{aesr}(\cdots \mathrm{aesr}(X \oplus K_0) \cdots \oplus K_{r-1}) \oplus K_r\,.$$

The tweakable blockcipher in AEZ is furthermore built of multiplications. Note that we can represent 128-bit strings as elements of a finite field $\mathrm{GF}(2^{128})$ of order 2^{128}, and vice versa: a 128-bit string $A = a_{127}a_{126}\cdots a_1a_0 \in \{0,1\}^{128}$ can be seen as a polynomial $A(\mathtt{x}) = a_{127}\mathtt{x}^{127} + \cdots a_1\mathtt{x} + a_0 \in \mathrm{GF}(2^{128})$. We define multiplication of $A, B \in \{0,1\}^{128}$ as multiplication in $\mathrm{GF}(2^{128})$ modulo the irreducible polynomial $f(\mathtt{x})$ used to generate the field:

$$A \cdot B := A(\mathtt{x}) \cdot B(\mathtt{x}) \bmod f(\mathtt{x})\,.$$

We remark that the multiplications in AEZ usually involve a term A of the form $2^m + n$ for $m \in \mathbb{N}$ and $n \in [0..7]$, which significantly simplifies the computation of $A \cdot B$. We refer to [17] for the details.

The tweakable blockcipher $\widetilde{E}$ of (2) takes as input a key $I\|J\|L \in \{0,1\}^{3\cdot128}$, a tweak $(j,i) \in \big(\{-1,0\} \times [0..7]\big) \cup \big(\mathbb{N}^+ \times \mathbb{N}\big)$, and a plaintext X and computes the ciphertext as

tweak	$\widetilde{E}^{j,i}_{I\|J\|L}(X) =$
$j = -1,\ i \in [0..7]$	$\mathrm{AES10}_{\mathbf{K}}(X)$ with $\mathbf{K} = (i \cdot J, I, J, L, I, J, L, I, J, L, I)$
$j = 0,\quad i \in [0..7]$	$\mathrm{AES4}_{\mathbf{K}}(X)$ with $\mathbf{K} = (i \cdot I, J, I, L, 0^{128})$
$j = 1,\quad i \in \mathbb{N}$	$\mathrm{AES4}_{\mathbf{K}}(X)$ with $\mathbf{K} = \big(\Delta_i \cdot I, J, I, L, 0^{128}\big)$
$j = 2,\quad i \in \mathbb{N}$	$\mathrm{AES4}_{\mathbf{K}}(X)$ with $\mathbf{K} = \big(\Delta_i \cdot I, L, I, J, L\big)$
$j \geq 3,\quad i = 0$	$\mathrm{AES4}_{\mathbf{K}}(X)$ with $\mathbf{K} = \big(2^{j-3} \cdot L, J, I, L, 2^{j-3} \cdot L\big)$
$j \geq 3,\quad i \geq 1$	$\mathrm{AES4}_{\mathbf{K}}(X)$ with $\mathbf{K} = \big(2^{j-3} \cdot L \oplus \Delta_i \cdot J, J, I, L, 2^{j-3} \cdot L \oplus \Delta_i \cdot J\big)$

where $\Delta_i = (2^{3+\lfloor (i-1)/8 \rfloor} + (i\text{–}1 \bmod 8))$ for brevity. This tweakable blockcipher reminds of the XE(X) tweakable blockcipher used in OCB2 [25], as the "inner keys" are invariant of the tweak, and the "outer keys" depend on the tweak via the powering-up methodology.

Hoang et al. [17] claim that the AEZ construction is secure as long as $\widetilde{E}$ is a secure tweakable blockcipher. The usage of the tweakable blockcipher $\widetilde{E}$ as described above is validated using the so-called proof-then-prune approach: first, it is argued that if the tweakable blockcipher is instantiated with AES10 everywhere, it behaves like XE(X), and then some uses of AES10 are cut down to 4 rounds to speed up AEZ. As it is unreasonable to assume that AES4 behaves like a pseudorandom permutation, the proof-then-prune approach is ultimately a heuristic [17, 20]. In this work, we will *not* consider any internal properties of AES4 and AES10, and simply consider both AES4 and AES10 as secure primitives: our attacks are independent of the debated proof-then-prune approach, but rather center around the structural properties of $\widetilde{E}$.

3.2 Weak Keys

The definition of $\widetilde{E}$, and more specifically the generation of the key $\mathbf{K}$ from $I\|J\|L$, reveals peculiar behavior. Particularly, if one of the subkeys I, J, L equals 0^{128}, the tweakable blockcipher allows for trivial collisions among different tweaks and is insecure.

Lemma 1. *The tweakable blockcipher $\widetilde{E}$ satisfies the following properties:*

(i) If $J = 0^{128}$, then $\widetilde{E}^{-1,i}_{I\|0^{128}\|L} = \widetilde{E}^{-1,i'}_{I\|0^{128}\|L}$ for any $i, i' \in [0..7]$;
(ii) If $I = 0^{128}$, then $\widetilde{E}^{0,i}_{0^{128}\|J\|L} = \widetilde{E}^{0,i'}_{0^{128}\|J\|L}$ for any $i, i' \in [0..7]$;
(iii) If $L = 0^{128}$, then $\widetilde{E}^{j,i}_{I\|J\|0^{128}} = \widetilde{E}^{j',i}_{I\|J\|0^{128}}$ for any $j, j' \geq 3$ and $i \in \mathbb{N}$.

Proof. The properties are in fact a direct consequence of the definition of $\widetilde{E}^{j,i}_{I\|J\|L}$ (see Section 3.1). Starting with (i): for subkey $J = 0^{128}$ and tweak value $j = -1$, we have

$$\widetilde{E}^{-1,i}_{I\|0^{128}\|L}(X) = \text{AES10}_{\mathbf{K}}(X)\,,$$

with $\mathbf{K} = (i \cdot 0^{128}, I, 0^{128}, L, I, 0^{128}, L, I, 0^{128}, L, I)$. In other words, $\widetilde{E}^{-1,i}_{I\|0^{128}\|L}(X)$ is independent of i, and we obtain that

$$\widetilde{E}^{-1,i}_{I\|0^{128}\|L} = \widetilde{E}^{-1,i'}_{I\|0^{128}\|L}$$

for any $i, i' \in [0..7]$. The proof of (ii) and (iii) is equivalent: for (ii), $\widetilde{E}^{0,i}_{0^{128}\|J\|L}$ is independent of i, and for (iii), $\widetilde{E}^{j,i}_{I\|J\|0^{128}}$ is independent of $j \geq 3$ for all $i \in \mathbb{N}$. □

More properties can be derived in a similar fashion, but these three relations suffice for the discussion of our attacks.

4 Weak Key Attacks on AEZ

We will perform three distinguishing attacks on AEZ, each of which exploits one of the properties of Lemma 1 and distinguishes AEZ from random in at most two queries. Note that if it is unknown which subkey equals 0^{128}, hence it is unknown which of the properties of Lemma 1 to exploit, all three attacks should be evaluated and the complexity is five queries (at most).

The first two distinguishing attacks rely on weaknesses in Encipher-AEZ-core, while the third one relies on a weakness in AEZ-hash. In these attacks, we will consider an adversary that has access to either $\mathcal{E}_K$ with random key K, or its idealized counterpart π (cf. Section 2.1), and denote by $\mathcal{O} \in \{\mathcal{E}_K, \pi\}$ the oracle to which the adversary has access.

4.1 Attack Exploiting Property (i)

Assume that $J = 0^{128}$. Using Lemma 1 property (i), we can perform the following distinguishing attack.

- Let N, A, τ be any nonce, associated data, and tag size;
- Let M be any message such that $384 \leq |M| + \tau < 511$. Write $M \| 0^\tau = M_\mathsf{u} \| M_\mathsf{v} \| M_\mathsf{x} \| M_\mathsf{y}$, where $|M_\mathsf{u}| = |M_\mathsf{x}| = |M_\mathsf{y}| = 128$ and $|M_\mathsf{v}| = |M| + \tau - 384 =: \ell$;
- Query $C = \mathcal{O}(N, A, \tau, M) \in \{0,1\}^{|M|+\tau}$. Write $C = C_\mathsf{u} \| C_\mathsf{v} \| C_\mathsf{x} \| C_\mathsf{y}$, where $|C_\mathsf{u}| = |C_\mathsf{x}| = |C_\mathsf{y}| = 128$, and $|C_\mathsf{v}| = \ell$;
- If
$$\mathrm{chop}_\ell\big(M_\mathsf{u} \oplus C_\mathsf{u} \oplus M_\mathsf{v} \oplus C_\mathsf{v}\big) = 0^\ell\,, \tag{3}$$
output 0, otherwise output 1.

If $\mathcal{O} = \mathcal{E}_K$, we have

$$\begin{aligned}\mathrm{chop}_\ell\big(M_\mathsf{u} \oplus C_\mathsf{u}\big) &= \mathrm{chop}_\ell\big(\widetilde{E}^{-1,4}_{I\|0^{128}\|L}(S)\big) \\ &\overset{\text{(i)}}{=} \mathrm{chop}_\ell\big(\widetilde{E}^{-1,5}_{I\|0^{128}\|L}(S)\big) = \mathrm{chop}_\ell\big(M_\mathsf{v} \oplus C_\mathsf{v}\big)\,,\end{aligned}$$

and (3) is satisfied by construction. Thus, the adversary always outputs 0 in the real world. In the ideal world, if $\mathcal{O} = \pi$, this condition is satisfied with probability $1/2^\ell$. Thus, the success probability of the attack is

$$\mathbf{Adv}^{\mathrm{rae}}_{\mathrm{AEZ}}(\mathcal{A}) = 1 - 1/2^\ell\,,$$

where $\mathcal{A}$ makes 1 construction query of length $|N|+|A|+|M|$, and has negligible time complexity. Recall that $\ell = |M| + \tau - 384$, where M is a freely chosen message. Hence, by taking $|M| + \tau = 511$ the success probability of the attack is $1 - 1/2^{127}$.

4.2 Attack Exploiting Property (ii)

Assume that $I = 0^{128}$. Using Lemma 1 property (ii), we can perform the following distinguishing attack.

- Let N, A, τ be any nonce, associated data, and tag size;
- Let $0 \leq \ell < 128$. Let $M_\mathsf{v}, M'_\mathsf{v} \in \{0,1\}^\ell$ be any two *distinct* message blocks. Write $M_\mathsf{u} = M_\mathsf{v} 10^*$ and $M'_\mathsf{u} = M'_\mathsf{v} 10^*$. Let $M_\mathsf{xy} \in \{0,1\}^{256-\tau}$ be any message block. Write
$$M = M_\mathsf{u} \| M_\mathsf{v} \| M_\mathsf{xy} \text{ and } M = M'_\mathsf{u} \| M'_\mathsf{v} \| M_\mathsf{xy}\;;$$
- Query $C = \mathcal{O}(N, A, \tau, M) \in \{0,1\}^{|M|+\tau}$ and $C' = \mathcal{O}(N, A, \tau, M') \in \{0,1\}^{|M'|+\tau}$. Write $C = C_\mathsf{u} \| C_\mathsf{v} \| C_\mathsf{x} \| C_\mathsf{y}$ and $C' = C'_\mathsf{u} \| C'_\mathsf{v} \| C'_\mathsf{x} \| C'_\mathsf{y}$, where $|C_\mathsf{u}| = |C_\mathsf{x}| = |C_\mathsf{y}| = |C'_\mathsf{u}| = |C'_\mathsf{x}| = |C'_\mathsf{y}| = 128$, and $|C_\mathsf{v}| = |C'_\mathsf{v}| = \ell$;

- If

$$M_{\mathsf{u}} \oplus C_{\mathsf{u}} \oplus M'_{\mathsf{u}} \oplus C'_{\mathsf{u}} = 0^{128}\,, \tag{4}$$

output 0, otherwise output 1.

The verification of the attack is a bit more complex than for case (i), and relies on the key observation that in the real world, $S = S'$. In more detail, if $\mathcal{O} = \mathcal{E}_K$, we have

$$X = X_{\mathsf{u}} \oplus X_{\mathsf{v}} = \widetilde{E}^{0,4}_{0^{128}\|J\|L}(M_{\mathsf{u}}) \oplus \widetilde{E}^{0,5}_{0^{128}\|J\|L}(M_{\mathsf{v}}10^*) \overset{\text{(ii)}}{=} 0^{128}\,, \text{ and}$$

$$X' = X'_{\mathsf{u}} \oplus X'_{\mathsf{v}} = \widetilde{E}^{0,4}_{0^{128}\|J\|L}(M'_{\mathsf{u}}) \oplus \widetilde{E}^{0,5}_{0^{128}\|J\|L}(M'_{\mathsf{v}}10^*) \overset{\text{(ii)}}{=} 0^{128}\,.$$

In other words, $X = X'$. Furthermore, as (τ, N, A) is the same in both evaluations,

$$\Delta = \text{AEZ-hash}(0^{128}\|J\|L, \tau, N, A) = \Delta'\,.$$

Finally, the two different queries satisfy $M_{\mathsf{xy}} = M'_{\mathsf{xy}}$. From Algorithm 2 we obtain that the intermediate value S is a function of M_{xy}, X, and Δ, and thus,

$$S = S'\,.$$

We consequently obtain

$$M_{\mathsf{u}} \oplus C_{\mathsf{u}} = \widetilde{E}^{-1,4}_{0^{128}\|J\|L}(S) = \widetilde{E}^{-1,4}_{0^{128}\|J\|L}(S') = M'_{\mathsf{u}} \oplus C'_{\mathsf{u}}\,,$$

and (4) is satisfied by construction. Thus, the adversary always outputs 0 in the real world. In the ideal world, if $\mathcal{O} = \pi$, this condition is satisfied with probability $1/2^{128}$. Thus, the success probability of the attack is

$$\mathbf{Adv}^{\text{rae}}_{\text{AEZ}}(\mathcal{A}) = 1 - 1/2^{128}\,,$$

where $\mathcal{A}$ makes 2 construction queries of length $|N|+|A|+|M|$, and has negligible time complexity.

4.3 Attack Exploiting Property (iii)

Assume that $L = 0^{128}$. Using Lemma 1 property (iii), we can perform the following distinguishing attack.

- Let τ be any tag size and M any message such that $384 \le |M| + \tau < 511$;[5]
- Let $N, N' \in \{0,1\}^{128}$ be any two *distinct* nonces;

[5] The condition on the message length is simply to assure that the attack goes via Encipher-AEZ-core of Algorithm 2. As a matter of fact, AEZ-prf and Encipher-AEZ-tiny use AEZ-hash in an identical way, and the attack applies equally well to messages of a different length.

– Query $C = \mathcal{O}(N, N', \tau, M) \in \{0,1\}^{|M|+\tau}$ and $C' = \mathcal{O}(N', N, \tau, M) \in \{0,1\}^{|M|+\tau}$;
– If

$$C \oplus C' = 0^{|M|+\tau}\,, \tag{5}$$

output 0, otherwise output 1.

If $\mathcal{O} = \mathcal{E}_K$, we have

$$\begin{aligned}\varDelta &= \text{AEZ-hash}(I\|J\|0^{128}, \tau, N, N') \\ &= \widetilde{E}^{3,1}_{I\|J\|0^{128}}(\langle\tau\rangle_{128}) \oplus \widetilde{E}^{4,1}_{I\|J\|0^{128}}(N) \oplus \widetilde{E}^{5,1}_{I\|J\|0^{128}}(N') \\ &\overset{\text{(iii)}}{=} \widetilde{E}^{3,1}_{I\|J\|0^{128}}(\langle\tau\rangle_{128}) \oplus \widetilde{E}^{4,1}_{I\|J\|0^{128}}(N') \oplus \widetilde{E}^{5,1}_{I\|J\|0^{128}}(N) \\ &= \text{AEZ-hash}(I\|J\|0^{128}, \tau, N', N) = \varDelta'\,.\end{aligned}$$

It follows from Algorithm 2 that $C = C'$, and that (5) is satisfied by construction. Thus, the adversary always outputs 0 in the real world. In the ideal world, if $\mathcal{O} = \pi$, this condition is satisfied with probability $1/2^{|M|+\tau}$. Thus, the success probability of the attack is

$$\mathbf{Adv}^{\text{rae}}_{\text{AEZ}}(\mathcal{A}) = 1 - 1/2^{|M|+\tau}\,,$$

where $\mathcal{A}$ makes 2 construction queries of length $256 + |M|$, and has negligible time complexity. Recall that τ can be freely chosen.

5 External Key Padding

We consider a specific scenario, called "external key padding," which shows the potential strength of the attacks of Section 4. Consider a user that uses AEZ as a black box. Instead of plugging his key K' into AEZ directly, he naively thinks speed-up could be achieved by artificially extending K' to a 384-bit key in advance:[6]

$$K \leftarrow K'\|0^{384-|K'|}\,.$$

Alternatively, one could consider a scenario where two users communicate, both set a part of the key, K'_a and K'_b, and the final key is established by padding in the middle:

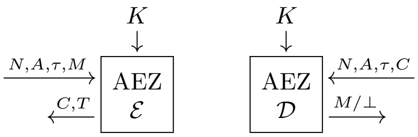
$$K \leftarrow K'_a\|0^{384-|K'_a|-|K'_b|}\|K'_b\,.$$

Although these use cases may sound contrived at first sight, they cover a realistic setting where a user of a scheme "misuses" it to suit the application. More generally, the scenario covers any form of poor key generation where $K \in \{0,1\}^{384}$

[6] Here, it is implicitly assumed that K' is of size at most 384 bits.

Use case of AEZ

- $K \leftarrow \{0,1\}^{3\cdot 128}$ derived using poor key generation
- Evaluation of AEZ $\mathcal{E}$ and $\mathcal{D}$:



- If K is weak, $\mathcal{E}$ can be distinguished from ideal π

Fig. 2: External key padding scenario. Here, the key K is generated in such a way that it may, inadvertently, contain a 0^{128} subkey.

is generated according to a very weak key generation function. This could happen due to naive use of the user, key generation regulations enforced by service providers, or whatsoever. See also Figure 2. Note that in these cases, AEZ works syntactically fine (as would any other authenticated encryption scheme) and will not produce errors due to the abuse.

5.1 How Does AEZ Behave?

It is straightforward to see that in case of external key padding, the attacks of Section 4 directly apply. Indeed, if a user of AEZ has a 256-bit key $K' \in \{0,1\}^{256}$, and prematurely pads it to a 384-bit key as $K = K' \| 0^{128}$, it obtains a weak key K for which property (iii) of Lemma 1 holds. Thus, the scheme can be broken in at most two queries, making use of the fact that the last subkey equals 0^{128}. A similar reasoning applies to the case $K \leftarrow K'_a \| 0^{128} \| K'_b$, where $K'_a, K'_b \in \{0,1\}^{128}$.

5.2 How Do Other Schemes Behave?

Intuitively, one would expect the security of the mode to decrease linearly with the amount of key reduction. In other words, if the security advantage as a function of the key size is $\mathcal{O}(2^{-|K|})$, then in case of the external key padding, the distinguishing advantage would increase to $\mathcal{O}(2^{-|K'|})$.

It turns out that, in fact, the majority of the authenticated encryption schemes show exactly this behavior. For instance, considering Sponge-based authenticated encryption [1,4,7,11,14,19,22,23,27], the key is already padded internally, and the external key padding has no influence. Alternatively, for regular blockcipher-based modes such as OCB [21], SIV [26], and COPA [3], the security of the mode is reduced to the security of the underlying E_K, and the adjustment from K' to K becomes captured in the blockcipher security $\mathbf{Adv}_E^{\mathrm{sprp}}(q,t)$.

6 Conclusion

Given the rarity of weak keys in AEZ (around 3 out of every 2^{128} keys), there is little chance that a randomly selected key is weak, and it seems not possible to break the security claims of AEZ using these weak keys. In addition, a simple mitigation of our attacks consists of imposing that no subkey equals 0^{128}. (But there may be other weak keys as multiplications in the tweakable blockcipher are performed in the finite field $\mathrm{GF}(2^{128})$ [16, 24, 28].)

Nevertheless, the observations *do* show a more peculiar weakness in AEZ, namely that the underlying tweakable blockcipher is not sound. Even if a more complicated key scheduling is used (as was done, for instance, in AEZ v2 and v3), it is still straightforward to see that a certain fraction of the keys allows for collisions in the tweakable blockcipher. In other words, while the issues with the external key padding could be mitigated using a stronger key scheduling, the issues with the tweakable blockcipher in AEZ are more structural.

Regardless of whether or not the external key padding scenario is relevant, it sets the stage for a discussion of what one may expect of a highly secure authenticated encryption scheme. Our observations (as well as the ones by Fuhr et al. [13] and Chaigneau and Gilbert [9]) stand in sharp contrast with the usage of powerful terms like "robustness" and with what high-security authenticated encryption embraces. Barwell et al. [6] already expressed their worries about the usage of the "robustness" term in the context of robust authenticated encryption, and stated: "Robustness characterizes the ability of a construct to be pushed right to the edge of its intended use case (and possibly beyond)." Putting our attacks in this perspective, by padding outside the mode, one incorrectly uses that mode, but on the other hand, "robust authenticated encryption" seems to imply that such modes work properly as long as they are employed in a *syntactically correct manner*. From this point of view, our attacks violate the robustness claims on AEZ.

Acknowledgments. Bart Mennink is supported by a postdoctoral fellowship from the Netherlands Organisation for Scientific Research (NWO) under Veni grant 016.Veni.173.017. The author would like to thank his COSIC colleagues, the attendees of Dagstuhl Symmetric Cryptography 2016, and the reviewers of CT-RSA 2017 for their comments and suggestions.

References

1. Andreeva, E., Bilgin, B., Bogdanov, A., Luykx, A., Mendel, F., Mennink, B., Mouha, N., Wang, Q., Yasuda, K.: PRIMATEs v1.02 (2015), submission to CAESAR competition
2. Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Mouha, N., Yasuda, K.: How to securely release unverified plaintext in authenticated encryption. In: Advances in Cryptology - ASIACRYPT 2014, Part I. Lecture Notes in Computer Science, vol. 8873, pp. 105–125. Springer, Heidelberg (2014)

3. Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Tischhauser, E., Yasuda, K.: Parallelizable and authenticated online ciphers. In: Advances in Cryptology - ASIACRYPT 2013, Part I. Lecture Notes in Computer Science, vol. 8269, pp. 424–443. Springer, Heidelberg (2013)
4. Aumasson, J., Jovanovic, P., Neves, S.: NORX v2.0 (2015), submission to CAESAR competition
5. Aumasson, J., Neves, S., Wilcox-O'Hearn, Z., Winnerlein, C.: BLAKE2: simpler, smaller, fast as MD5. In: Applied Cryptography and Network Security - ACNS 2013. Lecture Notes in Computer Science, vol. 7954, pp. 119–135. Springer, Heidelberg (2013)
6. Barwell, G., Page, D., Stam, M.: Rogue decryption failures: Reconciling AE robustness notions. In: IMA International Conference 2015. Lecture Notes in Computer Science, vol. 9496, pp. 94–111. Springer, Heidelberg (2015)
7. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Duplexing the sponge: Single-pass authenticated encryption and other applications. In: Selected Areas in Cryptography 2011. Lecture Notes in Computer Science, vol. 7118, pp. 320–337. Springer, Heidelberg (2011)
8. CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness (May 2014), `http://competitions.cr.yp.to/caesar.html`
9. Chaigneau, C., Gilbert, H.: Is AEZ v4.1 sufficiently resilient against key-recovery attacks? IACR Transactions on Symmetric Cryptology 1(1) (2016), to appear
10. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer (2002)
11. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon v1.1 (2015), submission to CAESAR competition
12. Espitau, T., Fouque, P.A., Karpman, P.: Higher-order differential meet-in-the-middle preimage attacks on SHA-1 and BLAKE. In: Advances in Cryptology - CRYPTO 2015, Part I. Lecture Notes in Computer Science, vol. 9215, pp. 683–701. Springer, Heidelberg (2015)
13. Fuhr, T., Leurent, G., Suder, V.: Collision attacks against CAESAR candidates - forgery and key-recovery against AEZ and Marble. In: Advances in Cryptology - ASIACRYPT 2015, Part II. Lecture Notes in Computer Science, vol. 9453, pp. 510–532. Springer, Heidelberg (2015)
14. Gligoroski, D., Mihajloska, H., Samardjiska, S., Jacobsen, H., El-Hadedy, M., Jensen, R.: π-Cipher v2.0 (2015), submission to CAESAR competition
15. Guo, J., Karpman, P., Nikolić, I., Wang, L., Wu, S.: Analysis of BLAKE2. In: CT-RSA 2014. Lecture Notes in Computer Science, vol. 8366, pp. 402–423. Springer, Heidelberg (2014)
16. Handschuh, H., Preneel, B.: Key-recovery attacks on universal hash function based MAC algorithms. In: Advances in Cryptology - CRYPTO 2008. Lecture Notes in Computer Science, vol. 5157, pp. 144–161. Springer, Heidelberg (2008)
17. Hoang, V.T., Krovetz, T., Rogaway, P.: AEZ v4: Authenticated Encryption by Enciphering (2015), submission to CAESAR competition
18. Hoang, V., Krovetz, T., Rogaway, P.: Robust authenticated-encryption AEZ and the problem that it solves. In: Advances in Cryptology - EUROCRYPT 2015, Part I. Lecture Notes in Computer Science, vol. 9056, pp. 15–44. Springer, Heidelberg (2015)
19. Jovanovic, P., Luykx, A., Mennink, B.: Beyond $2^{c/2}$ security in sponge-based authenticated encryption modes. In: Advances in Cryptology - ASIACRYPT 2014, Part I. Lecture Notes in Computer Science, vol. 8873, pp. 85–104. Springer, Heidelberg (2014)

20. Keliher, L., Sui, J.: Exact maximum expected differential and linear probability for 2-round Advanced Encryption Standard (AES). IET Information Security 1(2), 53–57 (2007)
21. Krovetz, T., Rogaway, P.: The software performance of authenticated-encryption modes. In: Fast Software Encryption 2011. Lecture Notes in Computer Science, vol. 6733, pp. 306–327. Springer, Heidelberg (2011)
22. Mennink, B., Reyhanitabar, R., Vizár, D.: Security of full-state keyed sponge and duplex: Applications to authenticated encryption. In: Advances in Cryptology - ASIACRYPT 2015, Part II. Lecture Notes in Computer Science, vol. 9453, pp. 465–489. Springer, Heidelberg (2015)
23. Morawiecki, P., Gaj, K., Homsirikamol, E., Matusiewicz, K., Pieprzyk, J., Rogawski, M., Srebrny, M., Wójcik, M.: ICEPOLE v2 (2015), submission to CAESAR competition
24. Procter, G., Cid, C.: On weak keys and forgery attacks against polynomial-based MAC schemes. In: Fast Software Encryption 2013. Lecture Notes in Computer Science, vol. 8424, pp. 287–304. Springer, Heidelberg (2013)
25. Rogaway, P.: Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In: Advances in Cryptology - ASIACRYPT 2004. Lecture Notes in Computer Science, vol. 3329, pp. 16–31. Springer, Heidelberg (2004)
26. Rogaway, P., Shrimpton, T.: A provable-security treatment of the key-wrap problem. In: Advances in Cryptology - EUROCRYPT 2006. Lecture Notes in Computer Science, vol. 4004, pp. 373–390. Springer, Heidelberg (2006)
27. Saarinen, M.: Beyond modes: Building a secure record protocol from a cryptographic sponge permutation. In: CT-RSA 2014. Lecture Notes in Computer Science, vol. 8366, pp. 270–285. Springer, Heidelberg (2014)
28. Saarinen, M.: Cycling attacks on GCM, GHASH and other polynomial MACs and hashes. In: Fast Software Encryption 2012. Lecture Notes in Computer Science, vol. 7549, pp. 216–225. Springer, Heidelberg (2012)