Breaking the confidentiality of OCB2

Bertram Poettering
Royal Holloway, University of London
bertram.poettering@rhul.ac.uk

2018-Nov-08

Abstract. OCB2 is a widely standardized mode of operation of a blockcipher that aims at providing authenticated encryption. A recent report by Inoue and Minematsu (IACR EPRINT report 2018/1040) indicates that OCB2 does not meet this goal. Concretely, by describing simple forging attacks the authors evidence that the (sub)goal of authenticity is not reached. The report does not question the confidentiality offered by OCB2.

In this note we show how the attacks of Inoue and Minematsu can be extended to also break the confidentiality of OCB2. We do this by constructing an IND-CCA adversary that requires minimal resources and achieves an overwhelming distinguishing advantage.

1 Introduction

In symmetric-key cryptography, a primitive providing authenticated encryption (AE) is one that allows for encrypting messages into ciphertexts, and decrypting ciphertexts into messages, such that both the confidentiality and the integrity of the messages are protected. A classic approach towards achieving this is through the hybrid encrypt-then-mac construction [1], but a line of research that started about two decades ago [3] and is now more active than ever [8] put forward several integrated AE modes that jointly achieve the two security goals in a more efficient way. As of today, authenticated encryption (possibly enriched with the option to take into account an associated-data string when performing the encryption and decryption operations; this variant is commonly referred to as AEAD) is a core component of many real-world cryptographic constructions.

While AES-GCM [4] has likely been the most widely used AE scheme for the last decade, a family of independent constructions is known by the name of OCB [5]. The three members of this family (OCB1, OCB2, OCB3) are blockcipher-based designs and effectively get along with a single blockcipher invocation per message block (‘rate-1’). While OCB1 is a plain AE mode, OCB2 and OCB3 are AEAD modes, and all three modes are among the most efficient (generic blockcipher based) designs that the market has to offer.[1]

In their recent report [2], Inoue and Minematsu (IM) showed that the OCB2 authenticated encryption scheme does not achieve the promised goal of authenticity. More precisely, the authors give four different attacks on OCB2 that allow for forging ciphertexts that validly decrypt to unauthentic messages without flagging an error. All four attacks succeed with overwhelming probability, they require minimal time and memory resources, and some of them allow for a specific level of control over the message to which a forged ciphertext will decrypt. The only realistic conclusion that one can draw from this seems to be that the authenticity of OCB2 is fully broken.[2]

Contribution. In this light it seems natural to ask whether the IM attacks on *authenticity* also have implications on the *confidentiality* of OCB2. In the abstract of their report, IM indicate that their attacks “do not break the privacy of OCB2” [2]. In the current article we thus provide a fresh IM-inspired assessment of the confidentiality of OCB2, and our main result is that the findings of IM indeed *can* be leveraged to yield effective message distinguishing attacks (in the IND-CCA sense). We show

[1] All versions of OCB were, and some still are, covered by intellectual property claims. This likely contributes to the clear real-world dominance of (the royalty-free) AES-GCM. The performance of AES-GCM could catch up with that of OCB only when CPU manufacturers started incorporating hardware support for certain GCM operations into their products (e.g., the `PCLMULQDQ` instruction in Westmere).

[2] As IM point out, it seems that fixing OCB2 is not too complicated. However, as ciphertexts of the original and the fixed version are not compatible with each other, implementers likely will, instead of applying the fix, take the opportunity to switch to AES-GCM in the first place.

this by giving concrete attacks that, just like the ones of IM, consume minimal resources and have an overwhelming success rate. Our conclusion is that the confidentiality of OCB2 is as broken as its authenticity.

2 Preliminaries

2.1 Notation

Symbols. If A is a set we write $a \leftarrow_{\$} A$ for the operation of picking an element of A uniformly at random and assigning it to the variable a. If B, B' are sets we write $B \overset{\cup}{\leftarrow} B'$ shorthand for $B \leftarrow B \cup B'$. For fixed-length strings $C_1, C_2 \in \{0,1\}^n$ we write $C \leftarrow C_1 \parallel C_2$ for their concatenation to a single ($2n$-long) string C. Such a string C can again be split up into its components C_1, C_2 with the $\overset{n}{\leftarrow}$ operator. For example $C \leftarrow C_1 \parallel C_2 \parallel C_3$ followed by $C'_1 \parallel C'_2 \parallel C'_3 \overset{n}{\leftarrow} C$ yields $C'_1 = C_1$, $C'_2 = C_2$, $C'_3 = C_3$. As further detailed in Section 2.3, we write $+$ for the xor operation. In program code we might use the if-then-else ternary operator `_ ? _ : _` known from programming languages like C and Java: If C is a Boolean condition, the expression $C \mathbin{?} a : b$ evaluates to a if C is true; otherwise, it evaluates to b.

Games. We define security notions via games played between a challenger and the adversary. Such games are formalized with pseudo-code. The execution of a game stops when it runs into a 'Stop' instruction. If the latter has an argument (e.g., 'Stop with x'), then the argument is considered the output of the game. For a game G we write $\Pr[G \Rightarrow 1]$ for the probability (over all random coins drawn by the game and the adversary) that the game terminates by running into a 'Stop with x' instruction with $x = 1$. We further use the instruction 'Require C', where C is a Boolean condition, as a shortcut for 'If $\neg C$: Stop with 0'. (This is usually used to penalize the adversary for posing 'illegal queries'; note that all our security definitions are such that such a penalty does not increase the formal attack advantage.)

2.2 Nonce-based AEAD

Syntax. We formalize a syntactical framework for authenticated encryption with associated data (AEAD). A corresponding scheme specifies a key space $\mathcal{K}$, a nonce space $\mathcal{N}$, an associated-data space $\mathcal{AD}$, a message space $\mathcal{M}$, a ciphertext space $\mathcal{C}$, and the (deterministic) algorithms enc and dec. The encryption algorithm enc takes a key $K \in \mathcal{K}$, a nonce $N \in \mathcal{N}$, an associated-data string $AD \in \mathcal{AD}$, and a message $M \in \mathcal{M}$, and outputs a ciphertext $C \in \mathcal{C}$. The decryption algorithm dec takes a key $K \in \mathcal{K}$, a nonce $N \in \mathcal{N}$, an associated-data string $AD \in \mathcal{AD}$, and a ciphertext $C \in \mathcal{C}$, and outputs either a message $M \in \mathcal{M}$ or the special symbol $\perp \notin \mathcal{M}$. If dec outputs a message, i.e., an element of $\mathcal{M}$, then we say it *accepts* (the ciphertext C); otherwise, if it outputs $\perp$, we say it *rejects*. For correctness we require that if keys, nonces, and associated data are provided consistently to enc and dec, then messages encrypted with enc are recovered by dec. Precisely, we require that for all $K \in \mathcal{K}, N \in \mathcal{N}, AD \in \mathcal{AD}, M \in \mathcal{M}$ we have $\mathsf{enc}(K, N, AD, M) = C \Rightarrow \mathsf{dec}(K, N, AD, C) = M$.

Security. We consider two security notions for AEAD: one for authenticity and one for confidentiality. While articles that aim at establishing the *security* of an AEAD candidate tend to do so using rather strong notions (e.g., in the IND\$ or SUF spirit), in this article we aim at analyzing the *insecurity* of an AEAD scheme and thus deliberately focus on rather weak notions. This only strengthens our results: If a scheme does not meet a weak notion, in particular it also does not meet any stronger notion. Note that the security goals that we formalize below assume nonce-respecting adversaries (that use for each encryption query a fresh nonce).

Our authenticity notion is formalized using the INT game from Figure 1 (left); the focus is on the integrity protection of associated-data strings and messages. (In contrast to [2] our definition disregards attacks that merely consist of manipulating nonces or ciphertexts.) We define the authenticity advantage of an adversary A as per $\mathbf{Adv}^{\mathrm{int}}(\mathsf{A}) := \Pr[\mathrm{INT}(\mathsf{A}) \Rightarrow 1]$. Intuitively, an AEAD scheme offers authenticity if $\mathbf{Adv}^{\mathrm{int}}(\mathsf{A})$ is negligible for all realistic adversaries A.

Our confidentiality notion is formalized using the IND^b games from Figure 1 (right); note that this is a classic left-or-right IND-CCA definition and thus captures a notion of confidentiality against active adversaries. (In contrast to [2] our definition does not require that ciphertexts look like random bitstrings.) We define the confidentiality advantage of an adversary B as per $\mathbf{Adv}^{\mathrm{ind}}(\mathsf{B}) := |\Pr[\mathrm{IND}^1(\mathsf{B}) \Rightarrow$

1] − Pr[IND^0(B) ⇒ 1]|. Intuitively, an AEAD scheme offers confidentiality (against active attacks) if $\mathbf{Adv}^{\mathrm{ind}}(\mathsf{B})$ is negligible for all realistic adversaries B.

```
Game INT(A)
00 NS ← ∅; ADM ← ∅
01 $K \leftarrow_{\$} \mathcal{K}$
02 $\mathsf{A}^{\mathcal{E}(\cdot,\cdot,\cdot),\mathcal{D}(\cdot,\cdot,\cdot)}$
03 Stop with 0

Oracle $\mathcal{E}(N, AD, M)$
04 Require $N \notin$ NS
05 $C \leftarrow \mathsf{enc}(K, N, AD, M)$
06 NS $\overset{\cup}{\leftarrow} \{N\}$
07 ADM $\overset{\cup}{\leftarrow} \{(AD, M)\}$
08 Return $C$

Oracle $\mathcal{D}(N, AD, C)$
09 $M \leftarrow \mathsf{dec}(K, N, AD, C)$
10 If $M = \bot$: Return $\bot$
11 If $(AD, M) \notin$ ADM:
12     Stop with 1
13 Return $M$

Game IND$^b$(B)
14 NS ← ∅; NADC ← ∅
15 $K \leftarrow_{\$} \mathcal{K}$
16 $b' \leftarrow \mathsf{B}^{\mathcal{E}(\cdot,\cdot,\cdot,\cdot),\mathcal{D}(\cdot,\cdot,\cdot)}$
17 Stop with $b'$

Oracle $\mathcal{E}(N, AD, M^0, M^1)$
18 Require $N \notin$ NS
19 Require $|M^0| = |M^1|$
20 $C \leftarrow \mathsf{enc}(K, N, AD, M^b)$
21 NS $\overset{\cup}{\leftarrow} \{N\}$
22 NADC $\overset{\cup}{\leftarrow} \{(N, AD, C)\}$
23 Return $C$

Oracle $\mathcal{D}(N, AD, C)$
24 Require $(N, AD, C) \notin$ NADC
25 $M \leftarrow \mathsf{dec}(K, N, AD, C)$
26 If $M = \bot$: Return $\bot$
27 Return $M$
```

Fig. 1. Games INT, IND^0, IND^1 for modeling integrity (of messages) and indistinguishability (of messages, under chosen-ciphertext attacks). Note how lines 04,06,18,21 enforce that the adversary be nonce-respecting.

2.3 Blockciphers that operate on finite fields

Blockciphers. For a key space $\mathcal{K}$ and a block length n, a blockcipher is a pair of functions $E, D \colon \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ such that for all $K \in \mathcal{K}$ and $X, Y \in \{0,1\}^n$ we have $D(K, E(K, X)) = X$ or, equivalently, $E(K, D(K, Y)) = Y$.

Finite Fields. The domain $\{0,1\}^n$ of a blockcipher can be identified with the set of elements of the finite field GF(2^n). More precisely, after fixing an irreducible degree-n polynomial $P \in \mathrm{GF}(2)[\mathtt{x}]$ (such a polynomial exists for all n), the elements of the field $\mathrm{GF}(2^n) := \mathrm{GF}(2)[\mathtt{x}]/(P)$ have a canonic representation as bitstrings of length n. We write + and · for the field operations (where + coincides with the xor operation). In the context of OCB2, the reduction polynomial P is chosen such that the field element $\mathtt{x}$ (the degree-1 monomial) is primitive in GF(2^n), that is, the sequence $\mathtt{x}^1, \mathtt{x}^2, \mathtt{x}^3, \ldots$ ranges over $2^n - 1$ different values.

2.4 Specification of OCB2

We reproduce details of the OCB2 nonce-based AEAD scheme from [6,7]. The scheme is based on a blockcipher (typically AES) and parameterized by a tag length τ (which kind of serves as a security parameter). In the following we actually do not give the full specification of OCB2; rather, in order to simplify the exposition, we remove some of its functionality (see upcoming paragraph). Note that any attack that is successful against the restricted scheme also applies to the full scheme. This holds in particular for the Inoue–Minematsu authenticity attacks from [2] as well as for the attack on confidentiality presented in the current article.

Assume a blockcipher (E, D) with key space $\mathcal{K}$ and block length n, and understand the cipher's domain $\{0,1\}^n$ as representing the elements of a finite field as suggested in Section 2.3. Then the algorithms of OCB2 are specified in Figure 2. The scheme has key space $\mathcal{K}$, nonce space $\{0,1\}^n$, and uses $\{0,1\}^*$ as associated-data space, message space, and ciphertext space. As announced above, our specification of OCB2 only covers a specific sub-case, namely the one where (a) the tag length coincides with the cipher's block length (this allows for disregarding the tag truncation step), (b) the associated-data input is always the empty string (this allows for removing the description of the auxiliary PMAC component), and (c) the length of considered messages and ciphertexts is always a multiple of the cipher's block length (this allows for neglecting padding operations). Note that the specifications of the enc and dec algorithms assume a length-encoding function $\mathrm{len}\colon \{0,1\}^{\leq n} \to \{0,1\}^n$.[3]

```
Algorithm $\mathsf{enc}_\tau(K, N, AD, M)$
00 Require $\tau = n \wedge AD = \epsilon \wedge n \mid |M|$
01 $L \leftarrow E(N)$
02 $M[1] \mathbin{\|} \ldots \mathbin{\|} M[m] \overset{n}{\leftarrow} M$
03 For $i \leftarrow 1$ to $m-1$:
04     $C[i] \leftarrow \mathtt{x}^i L + E(\mathtt{x}^i L + M[i])$
05 $C[m] \leftarrow M[m] + E(\mathtt{x}^m L + \mathrm{len}(0^n))$
06 $\Sigma \leftarrow M[1] + \ldots + M[m]$
07 $T \leftarrow E(\mathtt{x}^m(\mathtt{x}+1)L + \Sigma)$
08 $C \leftarrow C[1] \mathbin{\|} \ldots \mathbin{\|} C[m] \mathbin{\|} T$
09 Return $C$

Algorithm $\mathsf{dec}_\tau(K, N, AD, C)$
10 Require $\tau = n \wedge AD = \epsilon \wedge n \mid |C|$
11 $L \leftarrow E(N)$
12 $C[1] \mathbin{\|} \ldots \mathbin{\|} C[m] \mathbin{\|} T \overset{n}{\leftarrow} C$
13 For $i \leftarrow 1$ to $m-1$:
14     $M[i] \leftarrow \mathtt{x}^i L + D(\mathtt{x}^i L + C[i])$
15 $M[m] \leftarrow C[m] + E(\mathtt{x}^m L + \mathrm{len}(0^n))$
16 $\Sigma \leftarrow M[1] + \ldots + M[m]$
17 $T^* \leftarrow E(\mathtt{x}^m(\mathtt{x}+1)L + \Sigma)$
18 $M \leftarrow M[1] \mathbin{\|} \ldots \mathbin{\|} M[m]$
19 Return $(T = T^*)$ ? $M$ : $\perp$
```

Fig. 2. Specification of OCB2 (for the sub-case enforced by lines 00,10). For compactness we abbreviate $E(K,\cdot)$ with $E(\cdot)$ and $D(K,\cdot)$ with $D(\cdot)$.

3 Attacks on OCB2

We first recall a recent authenticity attack on OCB2 by Inoue and Minematsu (IM); then we derive from it an attack on the confidentiality of OCB2.

3.1 Inoue–Minematsu attack on authenticity

We reproduce the most simple attack on the authenticity of OCB2 from [2]. The attack gets along with a single encryption query and succeeds with finding a forgery with probability 1. (The delivery of the forgery requires, of course, an additional decryption query.) The details of a corresponding adversary A for the INT game from Figure 1 (left) are in Figure 3 (left). We trace the values of some variables throughout an execution of the adversary within the game:

$$\begin{aligned}
M[1] &= \mathrm{len}(0^n) \\
M[2] &= \mu \\
L_0 &= E(N_0) \\
C[1] &= \mathtt{x}^1 L_0 + E(\mathtt{x}^1 L_0 + M[1]) = \mathtt{x}^1 L_0 + E(\mathtt{x}^1 L_0 + \mathrm{len}(0^n)) \\
C[2] &= M[2] + E(\mathtt{x}^2 L_0 + \mathrm{len}(0^n)) = \mu + E(\mathtt{x}^2 L_0 + \mathrm{len}(0^n)) \\
\Sigma &= M[1] + M[2] = \mathrm{len}(0^n) + \mu \\
T &= E(\mathtt{x}^2(\mathtt{x}+1)L_0 + \Sigma) = E(\mathtt{x}^2(\mathtt{x}+1)L_0 + \mathrm{len}(0^n) + \mu) \\
C'[1] &= C[1] + \mathrm{len}(0^n) = \mathtt{x}^1 L_0 + E(\mathtt{x}^1 L_0 + \mathrm{len}(0^n)) + \mathrm{len}(0^n) \\
T' &= M[2] + C[2] = \mu + \mu + E(\mathtt{x}^2 L_0 + \mathrm{len}(0^n)) = E(\mathtt{x}^2 L_0 + \mathrm{len}(0^n)) \\
M'[1] &= C'[1] + E(\mathtt{x}^1 L_0 + \mathrm{len}(0^n)) = \mathtt{x}^1 L_0 + \mathrm{len}(0^n) \\
\Sigma' &= M'[1] = \mathtt{x}^1 L_0 + \mathrm{len}(0^n) \\
T^* &= E(\mathtt{x}^1(\mathtt{x}+1)L_0 + \Sigma') = E(\mathtt{x}^1(\mathtt{x}+1)L_0 + \mathtt{x}^1 L_0 + \mathrm{len}(0^n)) = E(\mathtt{x}^2 L_0 + \mathrm{len}(0^n)) \\
M' &= M'[1] = \mathtt{x}^1 L_0 + \mathrm{len}(0^n)
\end{aligned}$$

Note that by $T' = T^*$ the decryption oracle accepts ciphertext C' and returns the message $M' = \mathtt{x}^1 L_0 + \mathrm{len}(0^n)$. As M' and M have different lengths we in particular have $M' \neq M$ and the forgery counts. Thus $\mathbf{Adv}^{\mathrm{int}}(\mathsf{A}) = 1$, i.e., the adversary breaks authenticity with probability 1.

[3] The details of is function are specified in the OCB2 standard, but they are not relevant for our analysis.

```
Adversary $\mathsf{A}^{\mathcal{E}(\cdot,\cdot,\cdot),\mathcal{D}(\cdot,\cdot,\cdot)}$
00 Step 1:
01     Pick any $N_0 \in \{0,1\}^n$
02     Pick any $\mu \in \{0,1\}^n \setminus \{0^n\}$
03     $M[1] \leftarrow \mathrm{len}(0^n)$
04     $M[2] \leftarrow \mu$
05     $M \leftarrow M[1] \parallel M[2]$
06     Query $C \leftarrow \mathcal{E}(N_0, \epsilon, M)$
07     $C[1] \parallel C[2] \parallel T \overset{n}{\leftarrow} C$
08 Step 2:
09     $C'[1] \leftarrow C[1] + \mathrm{len}(0^n)$
10     $T' \leftarrow M[2] + C[2]$
11     $C' \leftarrow C'[1] \parallel T'$
12     Query $M' \leftarrow \mathcal{D}(N_0, \epsilon, C')$
13 Stop
```

```
Adversary $\mathsf{B}^{\mathcal{E}(\cdot,\cdot,\cdot,\cdot),\mathcal{D}(\cdot,\cdot,\cdot)}$
14 Steps 1+2 as in the IM attack,
15     obtaining $M' = \mathtt{x}^1 L_0 + \mathrm{len}(0^n)$
16 Step 3:
17     $L_0 \leftarrow \mathtt{x}^{-1}(M' + \mathrm{len}(0^n))$
18     Find pairs $(X_i, Y_i) \in E(K, \cdot)$
19     Pick $(N_1, L_1) \neq (N_0, L_0)$
20     $M^0[1] \leftarrow N_0 + \mathtt{x}^1(\mathtt{x}+1)L_1$
21     Pick any $M^1[1] \in \{0,1\}^n \setminus \{M^0[1]\}$
22     $M^0 \leftarrow M^0[1]$; $M^1 \leftarrow M^1[1]$
23     Query $C'' \leftarrow \mathcal{E}(N_1, \epsilon, M^0, M^1)$
24     $C''[1] \parallel T'' \overset{n}{\leftarrow} C''$
25     $b' \leftarrow (T'' = L_0)$ ? 0 : 1
26 Stop with $b'$
```

Fig. 3. Left: IM attack on authenticity from [2]. **Right:** Our new attack on confidentiality. See text for the meaning of lines 18,19.

We note that IM in [2] propose a total of four different attacks on the authenticity of OCB2, and here we reproduced just one of them. Other attacks from [2] generalize the one from Figure 3 (left) such that the message M assembled in line 05 is not anymore restricted to a single block; rather, multi-block messages of arbitrary length are allowed.

3.2 A novel attack on confidentiality

The IM attack from Figure 3 (left) breaks OCB2 by coming up with an unauthentic yet valid ciphertext C'. Perhaps surprisingly, the message M' corresponding to this ciphertext does not play a role in the attack; it is just discarded (line 12). In the following we show how the release of M' actually allows for conducting an attack on the confidentiality of OCB2. More precisely, after first emulating the steps of the IM attack to come up with ciphertext C', our confidentiality attacker uses the corresponding message M' to craft two challenge messages M^0, M^1 that can be distinguished within our left-or-right style security definition. In brief, the idea is to deduce from M' (and other public values) a set of ‘raw’ input-output pairs of $E(K, \cdot)$.[4] (Normal operation of OCB2 would discard unauthentic ciphertexts with the consequence that such pairs would remain hidden.) From these input-output pairs a fresh nonce N_1 and a message M^0 are derived such that at least one of the internal blockcipher invocations of operation $\mathsf{enc}(K, N_1, \epsilon, M^0)$ is on one of the known input values. This turns out to be sufficient for a distinguishing attack.

In Figure 3 (right) we provide the details of an adversary B for the IND^b games from Figure 1 (right). Attack steps 1 and 2 are just the ones from Figure 3 (left), where of course the original $\mathcal{E}(N_0, \epsilon, M)$ query (line 06) has to be replaced by the equivalent $\mathcal{E}(N_0, \epsilon, M, M)$ query. In particular adversary B obtains the message $M' = \mathtt{x}^1 L_0 + \mathrm{len}(0^n)$. This value immediately allows for deriving L_0. From the identities

$$\begin{aligned} C[1] &= \mathtt{x}^1 L_0 + E(\mathtt{x}^1 L_0 + \mathrm{len}(0^n)) \\ C[2] &= \mu + E(\mathtt{x}^2 L_0 + \mathrm{len}(0^n)) \\ T &= E(\mathtt{x}^2(\mathtt{x}+1)L_0 + \mathrm{len}(0^n) + \mu) \end{aligned}$$

that we established in Section 3.1, combined with the fact that all coefficients appearing in these equations are public values (with exception of the implicit blockcipher key K), we can derive three pairs $(X_i, Y_i) \in \{0,1\}^n \times \{0,1\}^n$ such that $E(K, X_i) = Y_i$.[5] Let $(N_1, L_1) = (X_i, Y_i)$ be one such pair, and assume w.l.o.g. that $(N_1, L_1) \neq (N_0, L_0)$.[6] Lines 18,19 in Figure 3 (right) implement these two steps. The remaining

[4] This step is also part of IM’s ‘Almost Universal Forgery, Variant 1’ attack [2, Sec 4.3].

[5] Concretely, $X_1 = \mathtt{x}^1 L_0 + \mathrm{len}(0^n)$, $Y_1 = C[1] + \mathtt{x}^1 L_0$, $X_2 = \mathtt{x}^2 L_0 + \mathrm{len}(0^n)$, $Y_2 = C[2] + \mu$, $X_3 = \mathtt{x}^2(\mathtt{x}+1)L_0 + \mathrm{len}(0^n) + \mu$, $Y_3 = T$.

[6] To see that this step is indeed w.l.o.g., observe that if $L_0 \neq 0^n$ then $X_1 \neq X_2$, and if $L_0 = 0^n$ then $X_2 \neq X_3$. In both cases one of X_1, X_2, X_3 will be different from N_0.

steps of our attack identify a message M^0 such that if M^0 is encrypted under N_1 then the tag-computing blockcipher invocation (line 07 in Figure 2) will be on input N_0, that is, the tag is arranged to be L_0; further, a second message M^1 is identified for which this is not the case. To analyze the success rate of our attack, observe that if B is executed in game IND^0 then the internal variables Σ'' and T'' of the encryption query in line 23 evaluate to

$$\begin{aligned} \Sigma'' &= M^0[1] = N_0 + \mathtt{x}^1(\mathtt{x}+1)L_1 \\ T'' &= E(\mathtt{x}^1(\mathtt{x}+1)L_1 + \Sigma'') = E(N_0) = L_0 \end{aligned}$$

and thus the adversary stops with output 0, while in game IND^1, as $E(K,\cdot)$ is a permutation, we have $T'' \neq L_0$ and the adversary stops with output 1. In any case the adversary is nonce-respecting and outputs $b' = b$, that is we have $\mathbf{Adv}^{\mathrm{ind}}(\mathsf{B}) = 1$ and the confidentiality of OCB2 is (fully) broken.

4 Conclusion

Extending the authenticity-focused findings of Inoue and Minematsu [2], we report on severe attacks on the confidentiality of OCB2. Our adversaries require little resources and achieve high success rates. While our attacks currently only target a formal security model (IND-CCA), it remains an open question whether our approach can also be used to attack OCB2 implementations in real-world environments. In any case, we believe that the results from both [2] and us allow for drawing only one conclusion: The OCB2 scheme is crucially broken, and all corresponding implementations should be upgraded as soon as possible (e.g., to AES-GCM [4] or a winner of the CAESAR competition [8]).

We note that IM in [2] also propose a fix for OCB2 that would rule out their attacks on authenticity. We believe that any such fix would also resolve the confidentiality issues that we identified.

References

1. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 531–545. Springer, Heidelberg, Germany, Kyoto, Japan (Dec 3–7, 2000)
2. Inoue, A., Minematsu, K.: Cryptanalysis of OCB2. Cryptology ePrint Archive, Report 2018/1040 (2018), https://eprint.iacr.org/2018/1040
3. Jutla, C.S.: Encryption modes with almost free message integrity. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 529–544. Springer, Heidelberg, Germany, Innsbruck, Austria (May 6–10, 2001)
4. McGrew, D.A., Viega, J.: The security and performance of the galois/counter mode of operation (full version). Cryptology ePrint Archive, Report 2004/193 (2004), http://eprint.iacr.org/2004/193
5. Rogaway, P.: OCB Mode. Personal website, http://web.cs.ucdavis.edu/~rogaway/ocb/
6. Rogaway, P.: Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 16–31. Springer, Heidelberg, Germany, Jeju Island, Korea (Dec 5–9, 2004)
7. Rogaway, P.: Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC (2004), http://web.cs.ucdavis.edu/~rogaway/papers/offsets.pdf
8. Various authors: CAESAR – Competition for Authenticated Encryption: Security, Applicability, and Robustness. Website (2018), https://competitions.cr.yp.to/caesar.html