Plaintext Recovery Attack of OCB2

Tetsu Iwata

Nagoya University, Japan
tetsu.iwata@nagoya-u.jp

Abstract. Inoue and Minematsu [Cryptology ePrint Archive: Report 2018/1040] presented efficient forgery attacks against OCB2, and Poettering [Cryptology ePrint Archive: Report 2018/1087] presented a distinguishing attack. In this short note, based on these results, we show a plaintext recovery attack against OCB2 in the chosen plaintext and ciphertext setting.

Keywords: OCB2, plaintext recovery attack, chosen plaintext and ciphertext setting

1 Introduction

OCB2 is an efficient authenticated encryption scheme proposed in [4]. In [1], Inoue and Minematsu presented efficient forgery attacks against OCB2, and Poettering presented an implication of the attack in the context of confidentiality [3]. Specifically, Poettering presented a distinguishing attack under the chosen plaintext and ciphertext setting, and this breaks the privacy security notion in the sense of IND-CCA.

In this short note, based on the results of [1,3], we show a plaintext recovery attack against OCB2 in the chosen plaintext and ciphertext setting, i.e., Poettering showed that OCB2 allows a distinguishing attack and hence cannot achieve the privacy notion of IND-CCA, but we show that for a given ciphertext, the corresponding plaintext can be recovered if the adversary has access to the encryption and decryption oracles. This attack model is usually not covered in the provable security analysis, as the provable security notion only requires less ambitious goal of the adversary (and this is an important point in provable security results). A similar model was considered in the security analysis of EAX-prime [2], while the attack against EAX-prime works without an encryption oracle.

To make the paper succinct, we follow exactly the same notation used in [1], and we omit the description of OCB2.

2 Plaintext Recovery Attack Model

We consider an attack model that closely follows [2]. A challenger has a secret key K. Let (C^*, T^*) be the encryption of (N^*, A^*, M^*), where a nonce N^*, associated

data A^*, and a plaintext M^* are arbitrarily chosen by the challenger. However, we make assumptions that M^* is long and C^* has many blocks (for instance 3 or more blocks), and that when C^* is broken into blocks as $(C^*[1], \ldots, C^*[m^*])$, there exist indices $j, k \in \{1, \ldots, m^* - 1\}$ such that $C^*[j] \neq C^*[k]$.

Then (N^*, A^*, C^*, T^*) is given to the adversary as a challenge. The adversary has access to the encryption and decryption oracles, and the goal is to recover M^*. The encryption oracle takes (N, A, M) as input and returns (C, T). The decryption oracle takes (N', A', C', T') as input and returns the corresponding plaintext M' or the reject symbol $\perp$.

The adversary cannot use N^* as a nonce in encryption queries (as N^* was already used in encryption to generate the challenge). Also, the adversary is nonce-respecting and hence cannot repeat the same nonce in encryption queries. To avoid a trivial win, the adversary cannot use the challenge (N^*, A^*, C^*, T^*) in decryption queries.

3 Plaintext Recovery Attack

Let (C^*, T^*) be the encryption of (N^*, A^*, M^*), and (N^*, A^*, C^*, T^*) is given to the adversary as a challenge. The goal is to recover M^*.

We first recover $L^* = E_K(N^*)$. For this, we first perform the attack in [1, Sect. 4.1].

1. Fix any $N, M[2] \in \{0,1\}^n$ such that $N \neq N^*$, and let $M = (M[1], M[2]) = (\mathtt{len}(0^n), M[2])$.
2. Next, make an encryption query (N, A, M), where A is empty, and obtain $(C[1], C[2], T)$.
3. Let $C' = C[1] \oplus \mathtt{len}(0^n)$ and $T' = M[2] \oplus C[2]$.
4. Make a decryption query (N', A', C', T'), where $N' = N$ and A' is empty, and obtain $M' = 2L \oplus \mathtt{len}(0^n)$.

As explained in [1, Sect. 4.3], the adversary can recover three input-output pairs $(X[1], Y[1]), (X[2], Y[2]), (X[3], Y[3])$ of the block cipher, where $Y[i] = E_K(X[i])$. If $X[i] = N^*$ for some i, then L^* is $Y[i]$. Otherwise, let (N'', L'') be one of $(X[i], Y[i])$ that satisfies $N'' \neq N$. We note that there is a negligible probability that this fails. However, this can be avoided by repeating the above Steps 1–4 with a different nonce N. We could also follow [1, Sect. 4.2] to obtain more input-output pairs. We then proceed as follows:

1. Fix any $A'' \in \{0,1\}^*$ and $M''[2] \in \{0,1\}^n$, and let $M'' = (M''[1], M''[2]) = (N^* \oplus 2L'', M''[2])$.
2. Make an encryption query (N'', A'', M'') and obtain $(C''[1], C''[2], T'')$.
3. Let L^* be $C''[1] \oplus 2L''$.

We see that Step 3 indeed gives L^*. See Fig. 1 for the generation process of $C''[1]$ for the encryption query (N'', A'', M'') of Step 2.

With the knowledge of L^*, following [1, Sect. 4.3], we modify C^* to make a decryption query. Specifically, let $C^* = (C^*[1], \ldots, C^*[m^*])$ be the challenge

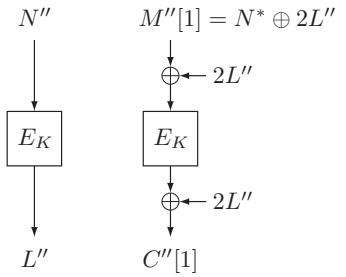

Fig. 1. The generation process of $C''[1]$ for the encryption query (N'', A'', M'') of Step 2. $L^* = E_K(N^*)$ is obtained as $C''[1] \oplus 2L''$.

ciphertext broken into blocks, and we first fix indices $j, k \in \{1, \ldots, m^* - 1\}$ that satisfy $C^*[j] \neq C^*[k]$. We then define $C^\$ = (C^\$[1], \ldots, C^\$[m^*])$ as follows:

- $C^\$[i] = C^*[i]$ for $i \in \{1, \ldots, m^*\} \setminus \{j, k\}$
- $C^\$[j] = C^*[k] \oplus 2^k L^* \oplus 2^j L^*$
- $C^\$[k] = C^*[j] \oplus 2^k L^* \oplus 2^j L^*$

Next, the adversary makes a decryption query $(N^*, A^*, C^\$, T^*)$, i.e, this is almost the same as the challenge, but the j-th and k-th blocks of C^* are modified. From the reasoning of [1, Sect. 4.3], the query will be accepted, and the adversary obtains $M^\$$. The goal of the attack, M^*, is obtained by swapping the j-th and k-th blocks of $M^\$$ and making necessary modifications. More precisely, from $M^\$ = (M^\$[1], \ldots, M^\$[m^*])$, we obtain $M^* = (M^*[1], \ldots, M^*[m^*])$ as follows:

- $M^*[i] = M^\$[i]$ for $i \in \{1, \ldots, m^*\} \setminus \{j, k\}$
- $M^*[j] = M^\$[k] \oplus 2^k L^* \oplus 2^j L^*$
- $M^*[k] = M^\$[j] \oplus 2^k L^* \oplus 2^j L^*$

See Fig. 2 for the encryption process of (N^*, A^*, M^*) and the decryption process of $(N^*, A^*, C^\$, T^*)$.

4 Conclusions

In this short note, based on the results of [1,3], we presented a plaintext recovery attack against OCB2 in the chosen plaintext and ciphertext setting. The distinguishing attack by Poettering [3] has already broken the confidentiality of OCB2 in the sense of IND-CCA. The result of this note shows that the confidentiality can also be broken in the sense of plaintext recovery.

Acknowledgements. The author would like to thank Kazuhiko Minematsu for useful comments.

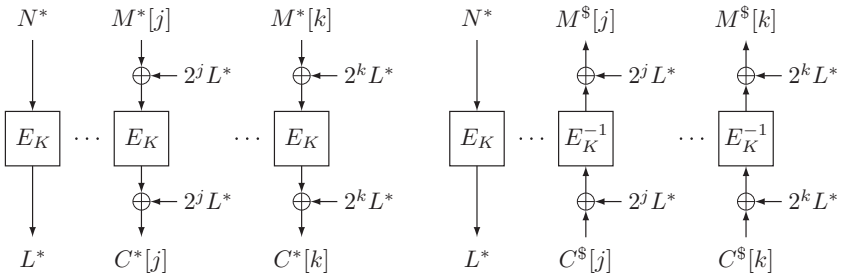

Fig. 2. The encryption process of (N^*, A^*, M^*) (left) and the decryption process of $(N^*, A^*, C^\$, T^*)$ (right). In the right figure, we have $C^\$[j] = C^*[k] \oplus 2^kL^* \oplus 2^jL^*$ and $C^\$[k] = C^*[j] \oplus 2^kL^* \oplus 2^jL^*$, and it follows that $M^*[j] = M^\$[k] \oplus 2^kL^* \oplus 2^jL^*$ and $M^*[k] = M^\$[j] \oplus 2^kL^* \oplus 2^jL^*$.

References

1. A. Inoue and K. Minematsu. Cryptanalysis of OCB2. *IACR Cryptology ePrint Archive*, 2018:1040, 2018.
2. K. Minematsu, S. Lucks, H. Morita, and T. Iwata. Attacks and Security Proofs of EAX-Prime. In S. Moriai, editor, *FSE 2013*, volume 8424 of *LNCS*, pages 327–347. Springer, 2013.
3. B. Poettering. Breaking the confidentiality of OCB2. *IACR Cryptology ePrint Archive*, 2018:1087, 2018.
4. P. Rogaway. Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In P. J. Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 16–31. Springer, 2004.