On the Security Claim of Tag Guessing of the AES-COPA Authenticated Encryption Algorithm

Jiqiang Lu

Infocomm Security Department, Institute for Infocomm Research,
Agency for Science, Technology and Research,
1 Fusionopolis Way, Singapore 138632
jlu@i2r.a-star.edu.sg

1 February 2015

Abstract. The COPA designers proved its integrity security to be (slightly below) the birthday bound [2, 3]. For AES-COPA [1], they claimed its integrity security is the birthday bound, and also claimed its security against tag guessing to be 128-bit without giving a detailed explanation. In this paper, we describe an (almost) universal forgery attack on AES-COPA in the nonce-respecting scenario, which requires nearly 2^{63} encryption queries with the total (associated data, message) pairs having a length of nearly 2^{64} blocks (which is very close to the approximate maximum length 2^{64} that AES-COPA can process with a single key), and a memory of about 2^{66} bytes, and has a time complexity of about 2^{62} memory accesses and a success probability of about 6%. We are not clear about their security definition on tag guessing; from a general understanding, it seems that our attack shows that this claim on the security against tag guessing for AES-COPA may be not correct.

Key words: Authenticated encryption algorithm, COPA, Universal forgery attack.

1 Introduction

COPA [2, 3] is a block-cipher-based authenticated encryption mode, which was proposed at ASIACRYPT '13 for parallel architectures such as general-purpose Central Processing Units and dedicated hardware. COPA was proved by the designers to have a birthday-bound security for its privacy and integrity, as long as the underlying block cipher is a strong pseudorandom permutation. In March 2014, the COPA instantiated with the AES block cipher under 128 key bits [1] (AES-COPA for short below) was submitted to the CAESAR competition [4] on authenticated encryption.

In this paper, we analyse the security of COPA against universal forgery attacks. We present a beyond-birthday-bound (almost) universal forgery attack

on COPA when used with variable associate data, following Fuhr et al.'s universal forgery attack [7] on Marble [8]. The attack has a data/memory/time complexity that is very near the birthday bound. When applied to AES-COPA, the attack requires nearly 2^{63} queries with the total (associated data, message) pairs having a length of nearly 2^{64} blocks (which is very close to the approximate maximum length 2^{64} that AES-COPA can process with a single key), and a memory of about 2^{66} bytes, and has a time complexity of about 2^{62} memory accesses and a success probability of about 6%. The attack can work in the nonce-respecting and nonce-misuse scenarios.

The designers claimed a 128-bit security against tag guessing for AES-COPA. We are not clear about their security definition; from a general understanding, it seems that our attack shows that this claim on the security against tag guessing for AES-COPA may be not correct.

2 Preliminaries

In this section, we give the notation used throughout this paper and briefly describe the COPA authenticated encryption algorithm.

2.1 Notation

We use the following notation.

$\oplus$	bitwise logical exclusive OR (XOR) operation
$*$	polynomial multiplication modulo the polynomial $x^{128} \oplus x^7 \oplus x^2 \oplus x \oplus 1$ in GF(2^{128})
e	the base of the natural logarithm ($\mathsf{e} = 2.71828\cdots$)

2.2 The COPA Authenticated Encryption Mode

The COPA [2, 3] authenticated encryption mode was published in 2013. Its internal state, key and tag have the same length. It has three phases: processing associate data, message encryption, and tag generation. Fig. 1 illustrates the message encryption and tag generation phase of COPA, where

- $\mathbf{E}_K$ is an n-bit block cipher with a k-bit user key K;
- $L = \mathbf{E}_K(0)$ is an n-bit secret internal parameter, which is called subkey sometimes [1];
- S is an n-bit internal state;
- $(AD_1, AD_2, \cdots, AD_{abn})$ is an associated data of abn n-bit blocks;
- $(M_1, M_2, \cdots, M_{mbn})$ is a message of mbn n-bit blocks;
- $(C_1, C_2, \cdots, C_{mbn})$ is the ciphertext for $(M_1, M_2, \cdots, M_{mbn})$; and
- T is the tag for $(M_1, M_2, \cdots, M_{mbn})$.

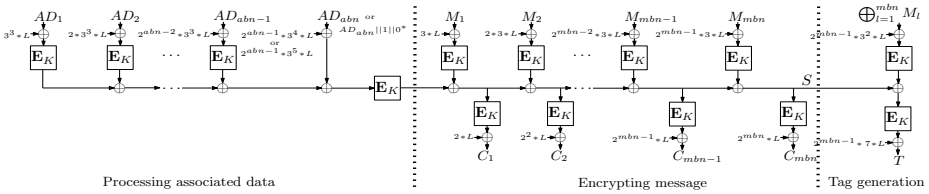

Fig. 1. Message encryption and tag generation of COPA

Decryption is the inverse of encryption, and tag verification is identical to tag generation. COPA can take no associate data, by setting the output of the processing associated data phase to zero. Please refer to [2,3] for the specification of COPA.

In 2014, an instantiation [1] of COPA that uses AES with 128 key bits [**?**] (i.e. AES-COPA) was submitted to the CAESAR competition [4], where a nonce is used and is appended to associate data, and the resulting value is treated as the associate data in the COPA mode.

We noted that the COPA designers did not distinguish between existential and universal forgeries in the specification of COPA [2, 3]; both were referred to be forgeries simply. But nevertheless, for AES-COPA in [1], they claimed its integrity security to be 64-bit according to the proved integrity security from [2, 3], and claimed its security against tag guessing to be 128-bit. There is no proof or explanation for the security claim against tag guessing.

3 Beyond-Birthday-Bound (Almost) Universal Forgery Attack on the COPA When Used with Variable Associated Data

We describe how to attack the COPA that uses variable associated data. The attack is based on Fuhr et al.'s universal forgery attack [7] on Marble.

3.1 Recovering the Secret Parameter L

The procedure is as follows, which is illustrated in Fig. 2.

1. Choose 2^η (associated data of one n-bit block long, fixed message of one n-bit block long) pairs $(AD_1^{(i)}, M_1) = (i, M_1)$, where $0 < \eta \leq \frac{n}{2}$ and $i = 0, 1, \cdots, 2^\eta - 1$. Query the COPA encryption and tag generation oracle, and obtain all the ciphertexts and tags for the 2^η (associated data, message) pairs; we denote by $C_1^{(i)}$ and $T^{(i)}$ the ciphertext and tag under associated data $AD_1^{(i)}$, respectively. Store $C_1^{(i)}$ into a table indexed by $C_1^{(i)}$.

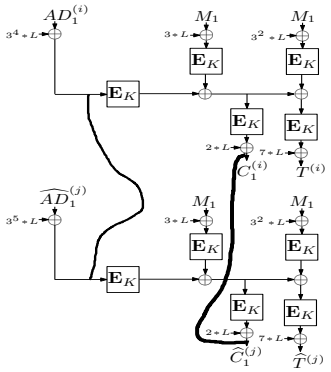

Fig. 2. State recovery attack on COPA

2. Choose $(2^\varphi - \rho)$ (associated data of less than n bits long, the same fixed message of one n-bit block long) pairs such that the (padded associated data, message) pairs $(\widehat{AD}_1^{(j)}, M_1) = (j \times 2^{\frac{n}{2}}, M_1)$, where $0 < \varphi \leq \frac{n}{2}$, $j = 1, 2, \cdots, 2^\varphi - 1$; if $\varphi = \frac{n}{2}$, then $j \neq 2^{\frac{n}{2}-1}$ and $\rho = 2$; and if $\varphi \neq \frac{n}{2}$, then $\rho = 1$. (The padded associated data are possible by the padding rule for associated data of COPA, namely, first a one then as many zeros as required to reach a multiple of the block size n. ρ represents the number of the impossible last blocks for padded associated data, that is 0 or 2^{n-1}.) Query the COPA encryption and tag generation oracle, and obtain all the ciphertexts and tags for the $(2^\varphi - \rho)$ (associated data, message) pairs; we denote by $\widehat{C}_1^{(j)}$ and $\widehat{T}^{(j)}$ the ciphertext and tag under associated data $\widehat{AD}_1^{(j)}$, respectively.
3. Check whether $\widehat{C}_1^{(j)}$ matches one of the set $\{C_1^{(i)} | i = 0, 1, \cdots, 2^\eta - 1\}$ for $j = 1, 2, \cdots, 2^\varphi - 1, j \neq 2^{\frac{n}{2}-1}$. We denote the match(es) by $(\widehat{C}_1^{(\omega)}, C_1^{(\mu)})$ if any, that is $\widehat{C}_1^{(\omega)} = C_1^{(\mu)}$.
4. For the match $(\widehat{C}_1^{(\omega)}, C_1^{(\mu)})$, we have $AD_1^{(\mu)} \oplus 3^4 * L = \widehat{AD}_1^{(\omega)} \oplus 3^5 * L$ by the structure of COPA. There, we can recover L from this equation.

The reason that we use padded associated data in Step 2 is that an input mask (i.e. $3^5 * L$) different from the one (i.e. $3^4 * L$) used in Step 1 will be introduced for the first block of (padded) associated data. This state recovery attack requires approximately $2^\eta + 2^\varphi$ encryption queries, a memory of approximately $n \cdot 2^\eta$ bits (as we do not need to store $\widehat{C}_1^{(j)}$), and has a time complexity of about 2^φ memory accesses (from Step 3) and a success probability of approximately $1 - \binom{2^\eta \cdot (2^\varphi - \rho)}{0} \cdot (2^{-n})^0 \cdot (1 - 2^{-n})^{2^\eta \cdot (2^\varphi - \rho)} \approx 1 - \mathsf{e}^{-2^{\eta+\varphi-n}}$.

3.2 Making an (Almost) Universal Forgery

If the secret parameter L is recovered by the above state recovery attack, we have two ways to make a universal forgery attack on COPA with a single query at a one-hundred-percent success probability. One way is based on modifying message, as follows. Assume a target (associated data of abn n-bit blocks long, message of mbn n-bit blocks long) pair $(AD, M) = (AD_1, AD_2, \cdots, AD_{abn}, M_1, M_2, \cdots, M_{mbn})$, where $abn \geq 0$ and $mbn > 0$.

1. Query the COPA encryption and tag generation oracle with the (associated data, message) pair $(AD, \widetilde{M}) = (AD_1, AD_2, \cdots, AD_{abn}, M_1, M_2, \cdots, M_{mbn}, 2^{mbn} * 3 * L \oplus 2^{mbn-1} * 3^2 * L \oplus \bigoplus_{i=1}^{mbn} M_i)$, and obtain its ciphertext $\widetilde{C} = (C_1, C_2, \cdots, C_{mbn}, \widetilde{C}_{mbn+1})$.
2. The ciphertext for (AD, M) is $C = (C_1, C_2, \cdots, C_{mbn})$, and the tag for (AD, M) is $\widetilde{C}_{mbn+1} \oplus 2^{mbn+1} * L \oplus 2^{mbn-1} * 7 * L$.

The other way is based on modifying associated data and is similar to Fuhr et al.'s universal forgery attack [7] on Marble, as follows. Assume a target (associated data of abn n-bit blocks long, message of mbn n-bit blocks long) pair $(AD, M) = (AD_1, AD_2, \cdots, AD_{abn}, M_1, M_2, \cdots, M_{mbn})$, where $abn > 0$ and $mbn \geq 0$.

1. Query the COPA encryption and tag generation oracle with the (associated data, message) pair $(\widetilde{AD}, M) = (AD_1, AD_2, \cdots, AD_{abn-1}, \widetilde{AD}_{abn}, \widetilde{AD}_{abn} \oplus 2^{abn} * 3^3 * L \oplus 2^{abn-1} * 3^3 * L, AD_{abn} \oplus 2^{abn-1} * 3^4 * L \oplus 2^{abn+1} * 3^4 * L, M_1, M_2, \cdots, M_{mbn})$, where $\widetilde{AD}_{abn}$ is an arbitrary block. Obtain its ciphertext and tag, denoted respectively by $\widetilde{C} = (\widetilde{C}_1, \widetilde{C}_2, \cdots, \widetilde{C}_{mbn})$ and $\widetilde{T}$.
2. The ciphertext for (AD, M) is $C = (\widetilde{C}_1, \widetilde{C}_2, \cdots, \widetilde{C}_{mbn})$, and the tag for (AD, M) is $\widetilde{T}$.

Particularly, when $\eta = \varphi = 64$ and $n = 128$, each universal forgery attack that includes the phase of recovering L requires approximately 2^{65} encryption queries, a memory of approximately 2^{68} bytes, and has a time complexity of 2^{64} memory accesses and a success probability of about 63%. (Note that if one would treat the time complexity for encrypting chosen messages as part of the time complexity of the attack, the resulting time complexity would be about $2^{65} \times 5 \approx 2^{67.4}$ block cipher encryptions.)

4 Application to AES-COPA

AES-COPA [1] has an additional (public) input parameter call nonce, which has a constant length of 128 bits. It is appended to associated data (if any), and then the resulting value is treated as associated data in COPA. As a consequence, when applying the above state recovery attack to AES-COPA, we should obtain associated data satisfying Steps 1 and 2; this can be easily done, for example, we choose (associated data of one 128-bit block long, nonce of one 128-bit long) pairs

$(AD, N^{(i)})$ in Step 1, and in Step 2 we choose the (associated data of less than 128 bits long, nonce of one 128-bit long) pairs such that the padded (associated data, nonce) pairs are $(AD, X^{(j)})$, where $N^{(i)} = AD_1^{(i)}$ and $X^{(j)} = \widehat{AD}_1^{(j)}$; and a value of AD can be $(1, \cdots, 1, 0)$ in binary form. Then, the first blocks for all the $(2^\eta + 2^\varphi - \rho)$ (padded) (associated data, nonce) pairs are identical, and the first block cipher encryption operations produce the same output, and we only need to modify the above state recovery attack slightly. As a result, the nonces used are different one another, and the state recovery attack works in the nonce-respecting scenario.

For AES-COPA, when we set $\eta = \varphi \approx 62$ extremely, the attack requires nearly 2^{63} queries with the total (associated data, message) pairs having a length of nearly 2^{64} blocks (which is very close to the approximate maximum length 2^{64} that AES-COPA can process with a single key), and a memory of about $2^{62} \times 16 = 2^{66}$ bytes, and has a time complexity of about 2^{62} memory accesses and a success probability of about 6%. (Note that for a longer (associated data, nonce, message) tuple, we need to reduce the values of η and φ accordingly.)

Remarks.

The designers claimed a 128-bit security against tag guessing for AES-COPA [1], (who claimed a 64-bit security on its integrity by the proved integrity security [2, 3]). We are not clear about how their security against tag guessing for AES-COPA is defined, and there is no proof or explanation for this security claim; anyway, from a general understanding of security against tag guessing, it seems that the above attack invalidates this security claim.

Observe that if there is a constraint on the maximum number of the blocks of an associated data or a message in COPA, the first attack described in Section 3.2 does not work for a message with the maximum number of blocks, and the second attack described in Section 3.2 does not work for an associate data with the number of blocks being two or one smaller than or equal to the maximum number. Thus, the attacks are almost universal forgery attacks [6]. Of course, we can combine the two ways together, so that the attacks can apply to a wider range of (associated data, message) pairs. (The attacks may apply to a message with the last block being an incomplete block.)

References

1. Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Tischhauser, E., Yasuda, K.: AES-COPA v.1, Submission to the CAESAR competition, March 2014. `http://competitions.cr.yp.to/round1/aescopav1.pdf`
2. Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Tischhauser, E., Yasuda, K.: Parallelizable and Authenticated Online Ciphers. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8269, pp. 424–443. Springer, Heidelberg (2013)
3. Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Tischhauser, E., Yasuda, K.: Parallelizable and Authenticated Online Ciphers. Cryptology ePrint Archive, Report 2013/790 (2013). `http://eprint.iacr.org/2013/790`

4. CAESAR — Competition for Authenticated Encryption: Security, Applicability, and Robustness. http://competitions.cr.yp.to/caesar.html
5. Daemen, J., Rijmen, V.: AES proposal: Rijndael. Presented at the First AES Candidate Conference. NIST, 1998.
6. Dunkelman, O., Keller, N., Shamir, A.: Almost universal forgery attacks on AES-based MAC's. Designs, Codes and Cryptography, avaialble as Online First. DOI: 10.1007/s10623-014-9969-x
7. Fuhr, T., Leurent, G., Suder, V.: Forgery and Key-Recovery Attacks on CAESAR Candidate Marble. HAL archive hal-01102031, 13 January 2015. http://hal.inria.fr/hal-01102031v2
8. Guo, J.: Marble Specification Version 1.1, Submission to the CAESAR competition, 26 March 2014. http://competitions.cr.yp.to/round1/marblev11.pdf