Observations on COMET

Daniel J. Bernstein[1,2], Henri Gilbert[3,4], and Meltem Sönmez Turan[5]

[1] Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607–7045, USA
[2] Horst Görtz Institute for IT Security, Ruhr University Bochum, Germany
djb@cr.yp.to
[3] ANSSI, Paris, France
[4] UVSQ, Versailles, France
henri.gilbert@ssi.gouv.fr
[5] Cryptographic Technology Group, National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg, MD 20899, USA
meltem.turan@nist.gov

Abstract. This note presents two attacks against COMET, a second-round candidate in the NIST lightweight cryptography standardization process. The first attack uses a long message to detect the use of weak keys, whereas the second attack focuses on the resistance of COMET against slide attacks. These attacks do not invalidate the security claims of the designers.

Keywords: COMET, distinguishers, lightweight cryptography, slide attacks, weak subkeys

1 Introduction

COMET (**CO**unter **M**ode **E**ncryption with authentication **T**ag), designed by Gueron et al. [1], is one of the second-round candidates in the NIST lightweight cryptography standardization process. COMET mode is parametrized by n, where $n \in \{64, 128\}$ is the block size of the underlying block cipher. Both COMET-64 and COMET-128 have key size $k = 128$, and COMET-128 is the primary variant of the candidate.

The main difference between COMET-64 and COMET-128 is the initialization process. The initial state of COMET can be viewed as an $(n+k)$-bit string $Y_0||Z_0$. Given the nonce N, the key K, and the underlying block cipher E, the initial state is defined as

Author list in alphabetical order; see https://www.ams.org/profession/leaders/culture/CultureStatement04.pdf. This work was supported by the U.S. National Science Foundation under grant 1913167, by the Cisco University Research Program, and by DFG Cluster of Excellence 2092 "CASA: Cyber Security in the Age of Large-Scale Adversaries". "Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation" (or other funding agencies). This work was initiated at a Schloss Dagstuhl seminar "Symmetric Cryptography". Permanent ID of this document: b15c58d84857ed7a7a13a6b29c1de525425da5de. Date: 2020.11.13.

- $Y_0 = K$ and $Z_0 = E_K(N)$, for COMET-128, and
- $Y_0 = E_K(0)$ and $Z_0 = K \oplus \mathrm{pad}(N)$, for COMET-64, where $\mathrm{pad}(N)$ is N followed by a zero byte; the nonce N has $k - 8$ bits for COMET-64.

The Z state consists of the k-bit subkey and is updated using a specific function φ, which is invertible and linear. The subkey has the form (L, R) where L and R each have $k/2$ bits. The update function φ does not affect the R part: $\varphi(L, R)$ has the form $(\ldots, R)$.

The COMET specification reports various attacks taking time 2^k for k-bit brute-force key search; or data 2^n for n-bit forgery; or time-data product 2^{n+k} for various other attacks; or time-data product $2^{n/2+k}$ for an attack using forward and backward queries to the block cipher. Khairallah [2] reports an attack that uses $2^{k/2}$ short known plaintexts, $2^{k/2}$ short forgery attempts, and time $2^{k/2}$.

In this note, we present two observations on COMET-64. Section 2 describes an attack that uses one chosen plaintext of length $2^{n/2}$ to detect whether a guess is correct for $k/2$ key bits. This leads to key recovery at cost $2^{n/2}$ with probability $2^{-k/2}$. This attack does not need any forgery attempts. Section 3 describes an attack that uses $2^{k/4+n/2}$ short chosen plaintexts, plus $2^{k/2}$ offline computations, for key recovery. This attack also does not need any forgery attempts. If $k = 128$ and $n = 64$ and data is limited to 2^{50} bytes then the success probability is about 2^{-38}. Attacking 2^{38} users would amplify the probability to about 1 but requires 2^{50} bytes of data per user. Note that the cost summaries ignore lower-order factors.

We presented our observations to the COMET designers in January 2020. They concurred with our analysis. Although the observations do not invalidate the security claims of the designers, to avoid the attack strategy the designers proposed a tweak to the key update function [3].

2 Colliding ciphertexts

Khairallah [2] defined a class of subkeys of the form $Z = (0^{k/2}, R)$ as weak. These subkeys are fixed points for the update function φ. Other subkeys have large period under φ. Khairallah also suggested a strategy to recognize a weak subkey using one short known plaintext and one forgery attempt.

In this section, we avoid forgeries and instead recognize weak subkeys using a long plaintext. Specifically, we encrypt a B-block message $(0^n, 0^n, \ldots, 0^n)$, where n is the block size of the underlying block cipher. Normally one would expect the B ciphertext blocks to be distinct with probability $(1 - 2^{-n})^{B(B-1)/2}$. This appears to be approximately correct if the initial subkey before message blocks is *not* weak. However, if the initial subkey *is* weak, then the ciphertext blocks are on a cycle of a permutation and are thus much more likely to be distinct. Note that the last block is encrypted differently but this makes negligible difference in the probabilities since there is only one such block.

We use this as a distinguisher between a weak initial subkey and non-weak initial subkeys. We take B somewhat larger than $2^{n/2}$ so that $(1-1/2^n)^{B(B-1)/2}$ is close to 0: e.g., if $n = 64$ and $B = 2^{35}$ then $(1 - 1/2^n)^{B(B-1)/2} < 2^{-46}$.

The above distinguisher applies to both COMET-64 and COMET-128. We now focus in particular on COMET-64. In this mode, the initial subkey before message blocks (here we assume for simplicity that the associated data is empty) is $K \oplus \mathrm{pad}(N) \oplus \mathtt{c}$, where K is the COMET key, N is a nonce, and $\mathtt{c}$ is $00||\mathtt{ctrl}_{\mathtt{pt}}||0^{k-5}$ as given in the COMET specification.

First, we guess the first $k/2$ bits of the secret key K. This guess, denoted G, is correct with probability $2^{-k/2}$. We then take any N such that $\mathrm{pad}(N) \oplus \mathtt{c}$ starts with G. Then $K \oplus \mathrm{pad}(N) \oplus \mathtt{c}$ results in a subkey that starts with $0^{k/2}$, i.e., the initial subkey is weak. We then apply the distinguisher above, encrypting a B-block message $(0^n, 0^n, \ldots, 0^n)$ and checking whether the output blocks are distinct. Of course, if B is large enough for this to be a reliable distinguisher, then we can afford to follow up by guessing the other key bits.

Compared to brute-forcing all $2^{k/2}$ keys that start with G, this attack costs only about $2^{n/2}$, which is better if n is much smaller than k, depending on the cost ratio between online queries and local computation.

3 Slide attacks

The observation presented in this section does not rely on the weak subkeys, and it would continue to apply even if keys were narrowed or nonce insertion were tweaked to avoid the weak subkeys.

Let M be the 10-block message $(0^n, \ldots, 0^n)$. First, we choose q distinct nonces $N_1, \ldots, N_q$ that agree after the first $k/2$ bits. For simplicity we assume that the rest of the nonce bits are chosen randomly. It might be better to take nonces in linear subspaces with as small intersection as possible.

Then, we encrypt M under each nonce using the secret key K. From each ciphertext $(C_0, C_1, \ldots, C_9)$ we extract seven 3-block subsequences: (C_0, C_1, C_2), (C_1, C_2, C_3), and so on through (C_6, C_7, C_8). We skip (C_7, C_8, C_9) since the last block of ciphertext is generated differently. We also ignore the tag attached to the ciphertext.

Next, we sort these $7q$ subsequences and look for collisions of the following forms:

- "slide distance 1": $(C_0, C_1, C_2) = (C'_1, C'_2, C'_3)$;
- "slide distance 2": $(C_0, C_1, C_2) = (C'_2, C'_3, C'_4)$;
- $\vdots$
- "slide distance 6": $(C_0, C_1, C_2) = (C'_6, C'_7, C'_8)$.

We have not analyzed whether 3 is the right choice of length of subsequence here.

One would expect that, for each choice of $(C_0, C_1, C_2, \ldots)$ and each choice of $(C'_0, C'_1, C'_2, \ldots)$, the equation $(C_0, C_1, C_2) = (C'_1, C'_2, C'_3)$ is satisfied with probability $1/2^{3n}$, and thus total probability $\leq q^2/2^{3n}$ across all q^2 pairs of queries. However, the slid-subkeys scenario described below produces these collisions.

We follow the COMET notation: subkey Z_{j+1} is used to encrypt an internal state producing a new state X_{j+1}, which is in turn used to encrypt message

block M_j (again assuming empty associated data) into ciphertext block C_j. The slide-distance-1-subkey scenario is, by definition, that $Z_1 = Z'_2$ and $X_1 = X'_2$.

This scenario produces the following effects. First, $C_0 = C'_1$. Furthermore, because all of our message blocks are identical and there is no other COMET state such as a counter, the next subkeys and states will satisfy $Z_2 = Z'_3$ and $X_2 = X'_3$, producing $C_1 = C'_2$. This situation persists until just before the last block of ciphertext, and in particular gives us $(C_0, C_1, C_2) = (C'_1, C'_2, C'_3)$. Similar comments apply to other slide distances.

To understand the probability that $Z_1 = Z'_2$, write $Z_1 = K \oplus \mathrm{pad}(N) \oplus \mathsf{c}$ as in the previous section, and write $Z'_1 = K \oplus \mathrm{pad}(N') \oplus \mathsf{c}$, where N and N' are the nonces for these messages. The next subkey Z'_2 is $\varphi(Z'_1) = \varphi(K) \oplus \varphi(\mathrm{pad}(N')) \oplus \varphi(c)$. Hence $Z_1 = Z'_2$ if and only if $K \oplus \mathrm{pad}(N) \oplus \mathsf{c} = \varphi(K) \oplus \varphi(\mathrm{pad}(N')) \oplus \varphi(\mathsf{c})$.

Recall that φ preserves the second half of its input: i.e., $\varphi(K)$ matches K on the last $k/2$ bits, $\varphi(\mathrm{pad}(N'))$ matches $\mathrm{pad}(N')$ on the last $k/2$ bits, and $\varphi(c)$ matches c on the last $k/2$ bits. Also $\mathrm{pad}(N')$ matches $\mathrm{pad}(N)$ on the last $k/2$ bits by our choices of nonces. Hence the last $k/2$ bits of $K \oplus \mathrm{pad}(N) \oplus c$ automatically match the last $k/2$ bits of $\varphi(K) \oplus \varphi(\mathrm{pad}(N')) \oplus \varphi(c)$.

On the first $k/2$ bits, our choices of N and N' are random, so

$$K \oplus \mathrm{pad}(N) \oplus c = \varphi(K) \oplus \varphi(\mathrm{pad}(N')) \oplus \varphi(c)$$

with probability $2^{-k/2}$. This does not imply that the probabilities are independent, but we heuristically assume that they are approximately independent.

We also heuristically assume that $X_1 = X'_2$ with probability 2^{-n}, independently of the condition $Z_1 = Z'_2$. In other words, we are in the slid-distance-1-subkey scenario with probability approximately $2^{-k/2-n}$.

There are q^2 of these collision opportunities, another q^2 for slide distance 2, etc., for a total probability approximately $1 - (1 - 2^{-k/2-n})^{6q^2}$. For small q the attack probability is approximately $\frac{6q^2}{2^{k/2+n}}$. The attack probability approaches 1 as q approaches the scale of $2^{k/4+n/2}$.

In the attack, if we see a collision $(C_0, C_1, C_2) = (C'_1, C'_2, C'_3)$, then we assume that we are in the slid-distance-1-subkey scenario, solve the equation $K \oplus \mathrm{pad}(N) \oplus \mathsf{c} = \varphi(K) \oplus \varphi(\mathrm{pad}(N')) \oplus \varphi(c)$ for the first $k/2$ bits of K, and brute-force the remaining bits. Similar comments apply to the other slide distances.

References

[1] Shay Gueron, Ashwin Jha, Mridul Nandi, *COMET: COunter Mode Encryption with authentication Tag* (2019). URL: https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/comet-spec-round2.pdf. Citations in this document: §1.

[2] Mustafa Khairallah, *Weak keys in the rekeying paradigm: application to COMET and mixFeed* (2019). URL: https://eprint.iacr.org/2019/888. Citations in this document: §1, §2.

[3] Shay Gueron, Ashwin Jha, Mridul Nandi, *Updates on COMET* (2020). URL: https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/comet_update.pdf. Citations in this document: §1.