A Simple Key-Recovery Attack on McOE-X

Florian Mendel[1,2], Bart Mennink[1], Vincent Rijmen[1], and Elmar Tischhauser[1]

[1] Katholieke Universiteit Leuven, ESAT/COSIC and IBBT, Belgium
[2] Graz University of Technology, IAIK, Austria

Abstract. In this paper, we present a key-recovery attack on the online authenticated encryption scheme McOE-X proposed by Fleischmann et al. at FSE 2012. The attack is based on the observation that in McOE-X the key is changed for every block of message that is encrypted in a deterministic way. This allows an adversary to recover the key by using a standard time-memory trade-off strategy. On its best setting the attack has a complexity as low as $2 \cdot 2^{n/2}$, while this should be 2^n for a good scheme. Taking AES-128 as an example this would result in an attack with complexity of 2^{65}.

Keywords: authenticated encryption, McOE-X, key-recovery attack.

1 Introduction

Motivation. Authenticated encryption is an important part in information security. Whenever two parties communicate over a network an authenticated encryption algorithm might be used to provide both privacy and authentication of the data. In most applications, there is not much value in keeping the data secret if they are not authenticated. Authentication of data is often of more value than their confidentiality.

Authenticated encryption can be generically constructed by combining an encryption scheme and a MAC. In [3], Bellare and Namprempre analyzed the three generic compositions of these two primitives: MAC-then-Encrypt (MtE), Encrypt-then-MAC (EtM), and Encrypt-and-MAC (E&M). They showed that the strongest notion of security for authenticated encryption can only be achieved by the EtM approach. However, schemes built from generic composition have some disadvantages. Besides that two different algorithms with two different keys are needed, the message needs to be processed twice, making the scheme impractical for some applications. Therefore, ISO/IEC specifies, next to the generic composition EtM, five authenticated encryption modes for block ciphers, namely OCB, SIV (Key Wrap), CCM, EAX, and GCM. Most of them are much faster than any solution which uses generic composition. All of them are proven to be secure against nonce-respecting adversaries assuming that the underlying block cipher is ideal. However, as pointed out in [8, 9] all these schemes, excluding SIV, are vulnerable to nonce-reusing adversaries. SIV has been explicitly designed to resist nonce-reuse attacks, but it has the disadvantage that it is inherently offline.

For encryption one must either keep the entire message in memory or read the message twice.

Therefore, Fleischmann et al. proposed a new family of authenticated encryption schemes in [8, 9] that are on the one hand secure against nonce-reusing adversaries and on the other hand are online. The construction extends the online encryption scheme TC3 by Rogaway and Zhang [16] to a provable secure nonce-reuse resistant online authenticated encryption scheme. The family consists of three members: McOE-X, McOE-D and McOE-G.

Our Contribution. In this paper, we present a key-recovery attack on McOE-X. The basic idea of the attack is very simple. Since in McOE-X the key is changed for every block of message that is encrypted, an adversary can recover the key by keeping the message input of some block cipher operation fixed and using a time-memory trade-off strategy. In its best setting the attack has a complexity as low as $2 \cdot 2^{n/2}$ with similar memory requirements, while this should be 2^n in the ideal case. Our attack allows a free trade-off between memory (precomputation) and time (online phase), and as such can be tailored to different attack scenarios. In all variants, it is significantly more efficient than Hellman's generic time-memory trade-off.

Note that this is close to the security bound of the McOE family. In more detail, Fleischmann et al. provide a formal security proof, which guarantees CCA3 indistinguishability up to about $2^{n/2}$. Since our key-recovery attack on McOE-X matches this bound (and from a theoretical point of view it even invalidates the proof), we took a detailed look at their security proof and identified a severe mistake that causes this gap: at a high level, Fleischmann et al. use ideal cipher results as if they were standard model results. These issues with the proof can, however, be resolved by explicitly considering the ideal cipher model.

Outline. The remainder of the paper is organized as follows. Section 2 describes the generic McOE construction and in particular McOE-X. In Section 3, we recall the security claims of the McOE construction respectively McOE-X. We present our key-recovery attack on McOE-X in Section 4 and discuss its relation to the security proof of McOE-X in Section 5. Finally, we discuss how McOE-X might be fixed in Section 6.

2 The McOE Family

The McOE construction is a new family of online authenticated encryption schemes recently proposed by Fleischmann et al. [8, 9]. It consists of three members: McOE-X, McOE-D and McOE-G. The general structure follows the online permutation approach described by Bellare et al. in [1] and is based on the Tweak Chain Hash construction [12] that is adapted from the Matyas-Meyer-Oseas construction. To be more precise, the construction itself is built on the online encryption scheme TC3 recently proposed by Rogaway and Zhang in [16]

that is based on a tweakable block cipher $\tilde{E}$. With an additional overhead of only two invocations of the tweakable block cipher $\tilde{E}$ the authors extend it to an online authenticated encryption scheme that is also secure against nonce-reusing adversaries. Both the encryption/authentication and the decryption/verification operations are described in Figure 1.

Encryption/Authentication $\mathcal{E}(K, V, M)$

1: *Partition* M *into* $M_1 \cdots M_L$
2: $U \leftarrow 0^n$
3: $\tau \leftarrow \tilde{E}(K, U, V)$
4: $U \leftarrow \tau \oplus V$
5: **for** $i = 1$ to L **do**
$C_i \leftarrow \tilde{E}(K, U, M_i)$
$U \leftarrow M_i \oplus C_i$
6: $T \leftarrow \tilde{E}(K, U, \tau)$
7: $C \leftarrow C_1 \cdots C_L$
8: **return** (C, T)

Decryption/Verification $\mathcal{D}(K, V, C, T)$

1: *Partition* C *into* $C_1 \cdots C_L$
2: $U \leftarrow 0^n$
3: $\tau \leftarrow \tilde{E}(K, U, V)$
4: $U \leftarrow \tau \oplus V$
5: **for** $i = 1$ to L **do**
$M_i \leftarrow \tilde{E}^{-1}(K, U, C_i)$
$U \leftarrow M_i \oplus C_i$
6: $T' \leftarrow \tilde{E}(K, U, \tau)$
7: $M \leftarrow M_1 \cdots M_m$
8: **if** $T = T'$ **return** M
else return FAIL

Fig. 1. Encryption/Authentication and Decryption/Verification operation of the McOE construction, where $\tilde{E}(K, U, \cdot)$ is a tweakable block cipher with key K and tweak U. Furthermore, M denotes the message, C denotes the ciphertext, V denotes the nonce and T is the authentication tag.

Additionally, Fleischmann et al. proposed a second scheme to provide length preservation using tag-splitting. The concept of tag-splitting is very similar to ciphertext stealing. We refer to the specification [9] for a detailed description of this method, since we do not need it for the attack described in this paper.

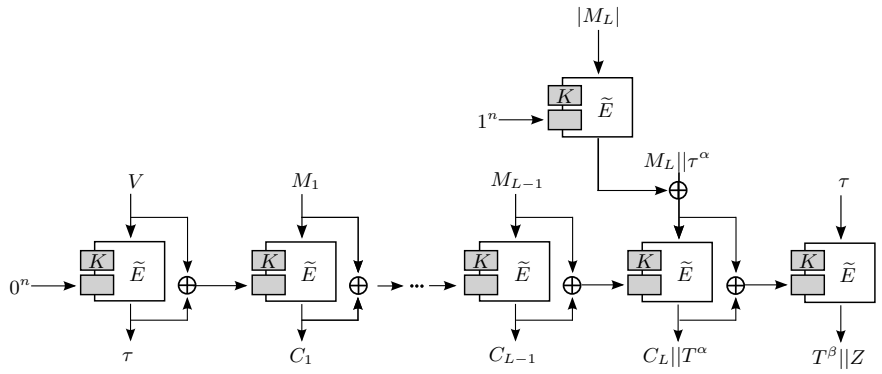
The generic construction of the McOE family with and without tag-splitting is depicted in Figure 2, where $\tilde{E}$ denotes a n-bit tweakable block cipher and V is a nonce. Due to the current lack of a dedicated n-bit block cipher with an n-bit tweak, Fleischmann et al. proposed three different constructions to convert an ordinary block cipher into a tweakable block cipher resulting in the three members of the McOE family: McOE-X, McOE-D and McOE-G.

2.1 McOE-X

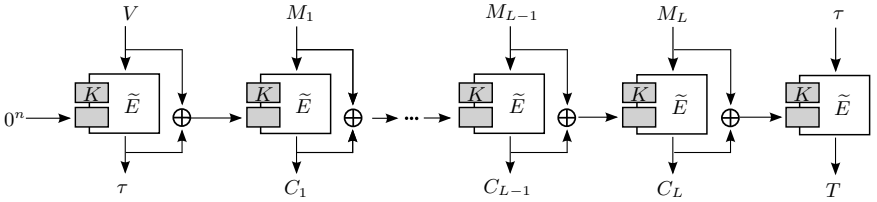
In this instance of the McOE family the tweak U (i.e. the chaining value) is xored to the key K to turn the block cipher E into a tweakable block cipher

$$\tilde{E}(K, U, \cdot) := E(K \oplus U, \cdot) \tag{1}$$

As noted by the designers for McOE-X related-key security is needed for the block cipher E. However, this requirement is not needed for the other two instances of the McOE family.



(a) with tag-splitting



(b) without tag-splitting

Fig. 2. Outline of the generic McOE construction (a) with and (b) without tag-splitting [9].

2.2 Other Members of the McOE Family

In addition to McOE-X, Fleischmann et al. proposed two other members of the McOE family not requiring related-key security, McOE-D and McOE-G. The first uses two block cipher invocations per message block to update the chaining value similar to the TCH-CBC construction as described in [5]. The second is based on the HCBC2 construction described in [2] and uses a universal hash function to update the chaining value. For a detailed description of the two schemes we refer to the specification [9].

3 Security of the Schemes

Fleischmann et al. [9] analyze their McOE schemes with respect to CCA3 security, a security notion for authenticated cryptosystems proposed in [15]. We informally describe the security definitions, referring to [9] for a more formal treatment. For an authenticated cryptosystem $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, where $\mathcal{K}$ denotes the key derivation function, we denote by $\mathbf{Adv}_{\Pi}^{\mathrm{CCA3}}(q, \ell, t)$ the CCA3 security of Π against any nonce-reusing adversary A, where q denotes the number of total queries an adversary A is allowed to ask to $\mathcal{E}$ and $\mathcal{D}$, ℓ the total length in blocks of the queries, and t the running time of A.

They derive the following result for the McOE schemes without tag-splitting.

Theorem 1 (Thm. 2 of [9]). *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a McOE scheme based on a tweakable block cipher $\tilde{E}$. Then,*

$$\boldsymbol{Adv}_{\Pi}^{CCA^3}(q, \ell, t) \leq \frac{3(q+\ell)(q+\ell+1) + 4q + 3\ell}{2^n - (q+\ell)} + 3\delta.$$

Here, $\delta = \boldsymbol{Adv}_{\tilde{E}}^{IND}(q+\ell, O(t))$ denotes the advantage of distinguishing $\tilde{E}$ from an ideal tweakable block cipher, where $q+\ell$ denotes the number of queries an adversary A is allowed to ask to $\tilde{E}$ and $\tilde{E}^{-1}$ and $O(t)$ the running time of A.

A significantly worse bound is obtained for McOE with tag-splitting. We refer to [9] for more details.

For the McOE-X construction without tag-spitting, they generalize this result as follows. Note that in this case δ refers to the related-key advantage of distinguishing E from an ideal block cipher.

Theorem 2 (Thm. 4 of [9]). *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a McOE-X scheme based on a block cipher E. Then,*

$$\boldsymbol{Adv}_{\Pi}^{CCA^3}(q, \ell, t) \leq \frac{3(q+\ell)(q+\ell+1) + 4q + 3\ell}{2^n - (q+\ell)} + 3\delta.$$

Here, $\delta = \boldsymbol{Adv}_{E}^{RK}(2q+\ell, O(t))$ denotes the related-key advantage of distinguishing E from an ideal block cipher, where $2q+\ell$ denotes the number of queries an adversary A is allowed to ask to E and E^{-1} and $O(t)$ the running time of A.

Again, the bound is slightly worse in case of tag-splitting and we refer to [9] for more details.

4 Our Attack on McOE-X

In this section, we propose our simple key-recovery attack on McOE-X. The attack consists of two phases: an offline (precomputation) phase and an online phase. It is a chosen plaintext attack and in its best setting it has a complexity as low as $2 \cdot 2^{n/2}$ with similar memory requirements.

4.1 Basic Attack

The basic idea of our attack can be explained as follows. The McOE-X mode changes the key for every block of plaintext that is encrypted. By keeping the plaintext input of some block cipher operation fixed, the adversary can exploit a basic time-memory trade-off strategy.

Let $E(k, x)$ denote the raw block cipher encryption operation with key k and plaintext x, and denote by K the target key we want to recover. Since the nonce plays no role in our attack, we omit it from the notation. The attack goes as follows.

Offline Phase (Precomputation)

1. Choose an arbitrary value a.
2. Repeat r times:
   (a) Choose a new value for k.
   (b) Compute $b = E(k, a)$ and save the pair (b, k) in a list L_1.

Online Phase

Repeat $2^n/r$ times:

1. Choose a new value for x and set $m = x\|a$.
2. Ask for the ciphertext/tag pair $(c, T) = \text{McOE-X}(K, m)$, with $c = C_1\|C_2$, and save the pair $(x \oplus C_1, C_2)$ in a list L_2.

Every match between a b-value in the list L_1 and a C_2-value in the list L_2 gives a candidate key $K = k \oplus x \oplus C_1$. We have set the number of iterations such that the expected number of matches between the two lists equals 1. Since the expected number of false alarms is small, we can state that the algorithm finds the correct key with a total complexity of approximately $r + 2^n/r$ encryptions.

Note that the queries in the online phase can be grouped. The adversary can ask for the encryption of $m = x\|a\|a\|\ldots\|a$ and save in L_2 the pairs $(x \oplus C_1, C_2), (a \oplus C_2, C_3), (a \oplus C_3, C_4), \ldots$. In this way the total number of block cipher encryptions is reduced.

Obviously by choosing $r = 2^{n/2}$ the attack has the best overall complexity, considering both the offline and the online phase, resulting in a final attack complexity of about $2 \cdot 2^{n/2}$ and similar memory requirements. We want to note that in the online phase of the attack we do not need to store the values in a list L_2 which reduces the memory requirements of the attack.

Sometimes an attacker wants to recover more than only a single key. In these cases only the second phase of the attack has to be repeated, while the precomputation phase has to be done only once. In such settings, in particular if the number of attacked keys is large, other values of r might result in a better overall complexity. In Table 1 we give the complexities and memory requirements for different choices of r.

Table 1. Complexities and memory requirements for both phases of the attack with different choices of r.

$\log_2(r)$	offline phase	online phase	memory	total
$n/4$	$2^{n/4}$	$2^{3n/4}$	$2^{n/4}$	$2^{3n/4}$
$n/3$	$2^{n/3}$	$2^{2n/3}$	$2^{n/3}$	$2^{2n/3}$
$n/2$	$2^{n/2}$	$2^{n/2}$	$2^{n/2}$	$2 \cdot 2^{n/2}$
$2n/3$	$2^{2n/3}$	$2^{n/3}$	$2^{2n/3}$	$2^{2n/3}$
$3n/4$	$2^{3n/4}$	$2^{n/4}$	$2^{3n/4}$	$2^{3n/4}$

4.2 A Memory-less Variant of the Attack

In practice, there is a profound imbalance between the cost of storage and the cost of computations. Hence, the high memory requirements of the attack could be seen as the bottleneck of the attack. It is therefore important to note that the attack with $r = 2^{n/2}$ can be implemented with negligible memory requirements and only a small increase in runtime by using a memory-less variant of the meet-in-the-middle attack introduced by Quisquater and Delescaille [14].

4.3 Comparison to Hellman's TMTO Attack

In [10] Hellman described a generic cryptanalytic TMTO attack on DES. Even though the attack was specifically designed for DES, it is applicable to any block cipher. For a block cipher with a key size of n bits and a precomputation with time complexity of about 2^n, Hellman's method has an (online) time complexity of $T = 2^{2n/3}$ and memory requirements of $M = 2^{2n/3}$. In more detail, it allows a time/memory trade-off curve of $M \cdot \sqrt{T} = 2^n$. Since we are only interested in attacks with $T \leq 2^n$ (faster than brute force), M has to be at least $2^{n/2}$. We want to note that the attack described in this paper is on a much better time/memory trade-off curve, i.e. $M \cdot T = 2^n$, and does not require a 2^n precomputation.

5 Relation of the Attack to the Security Proof

Fleischmann et al. [9] derive a security proof for McOE, which they also generalize to McOE-X. They derive security up to approximately $2^{n/2}$ queries (see Thm. 2). Although we want to stress that our attack is a key-recovery attack, which is much stronger than a distinguishability attack, it does not seem to directly invalidate the security proof of [9]. Yet, it turns out to expose a critical weakness in the security proof.

In short, the proof is technically invalid due to the fact that the authors (implicitly) consider security in the standard model. The security advantage is expressed in terms of parameters q, ℓ, and t. A critical observation is that q *only* denotes the number of queries made by the adversary to the full evaluation of McOE-X ($\mathcal{E}$ or $\mathcal{D}$), and in fact, the queries made in the offline phase of our attack *do not count* as queries. The adversary is considered to have free access to the underlying block cipher E and this offline phase only influences the variable t.

In this respect, for our attack we have parameters $q = 2^n/r$, $\ell = 2 \cdot 2^n/r$, and $t \approx r + 2 \cdot 2^n/r$. Now, considering the security claims of [9] for McOE-X in more detail (see Thm. 2 of this work), we see that the first part of the bound is independent of t.[3] As the authors claim, this part of the bound is determined by the event that a collision for the keyed compression function $f(K, U, M) = E(K \oplus U, M) \oplus M$ occurs, and the bound is obtained by applying the results of Black et al. [6, 7] for the PGV compression functions [13]. However,

[3] When we apply our attack for $r = 2^n$, this part of the bound misleadingly suggests an almost zero advantage.

the authors oversee that these results *do not apply*: Fleischmann et al. consider the standard model where the underlying block cipher E is freely accessible by the adversary, while the results of Black et al. hold in the ideal model, where E is an idealized block cipher to which the adversary has query access only. Note that in our attack, the success probability of a collision (between L_1 and L_2) increases if more offline computation is done: we could for example choose $r = 2^{3n/4}$ and recover the key with $q = 2^{n/4}$ queries (see Table 1). In fact, (contrived) examples are known where the results of Black et al. do not apply when the PGV compression functions are instantiated with a CCA secure block cipher [11]. In [4], Biryukov et al. present an attack on the Davies-Meyer compression function $f(U, M) = E(M, U) \oplus U$ when instantiated with the AES block cipher.

In order to restore the proof of Thm. 1 as given in [9], one needs to consider the ideal cipher model for $\tilde{E}$. This means that an adversary has query access to $\tilde{E}$ and $\tilde{E}^{-1}$ (next to the query access to $\mathcal{E}$ and $\mathcal{D}$). In this way, the results of Black et al. *do* apply. Additionally, the second part of the bound of Thm. 1 gets superfluous: an ideal cipher is obviously perfectly indistinguishable from an ideal cipher, and hence $\delta = 0$. The same remarks apply to Thm. 2. Note that in this model, the evaluations in the first phase of our attack are counted as queries, and the attack corresponds to parameters $q = r + 2^n/r$ and $\ell = r + 2 \cdot 2^n/r$.

6 How to Fix McOE-X

As an alternative to McOE-X one can always use McOE-D or McOE-G which are not vulnerable to the attack presented in this paper. However, both constructions have some drawbacks. In McOE-D two block cipher invocations are needed per message block processed and in McOE-G a universal hash function is used to update the chaining value.

The main problem of McOE-X construction is that the tweak U (i.e. the chaining value) of n bits is xored to the key K of also n bits to turn the block cipher E into a tweakable block cipher. This allows generic TMTO attacks on the construction with complexity as low as $2 \cdot 2^{n/2}$ in its best setting as described in Section 4. For instance in the case of AES-128 this could be as low as 2^{65}. One option to fix the construction with still using only a single block cipher invocations per message block processed is to use a block cipher with a key input of $2n$ bits instead of n bits.

$$\tilde{E}(K, U, \cdot) := E(K \| U, \cdot) \tag{2}$$

For instance AES-256 seems to be natural choice and the performance overhead compared to AES-128 is not so large, only about 40%.

Acknowledgments. This work was supported in part by the IAP Programme P6/26 BCRYPT of the Belgian State (Belgian Science Policy) and by the European Commission through the ICT programme under contract ICT-2007-216676 ECRYPT II. In addition, this work was supported by the Research Fund KU Leuven, OT/08/027.

References

1. Mihir Bellare, Alexandra Boldyreva, Lars R. Knudsen, and Chanathip Namprempre. Online Ciphers and the Hash-CBC Construction. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *LNCS*, pages 292–309. Springer, 2001.
2. Mihir Bellare, Alexandra Boldyreva, Lars R. Knudsen, and Chanathip Namprempre. On-Line Ciphers and the Hash-CBC Constructions. Cryptology ePrint Archive, Report 2007/197, 2007.
3. Mihir Bellare and Chanathip Namprempre. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. *J. Cryptology*, 21(4):469–491, 2008.
4. Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolić. Distinguisher and Related-Key Attack on the Full AES-256. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *LNCS*, pages 231–249. Springer, 2009.
5. John Black, Martin Cochran, and Thomas Shrimpton. On the Impossibility of Highly-Efficient Blockcipher-Based Hash Functions. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *LNCS*, pages 526–541. Springer, 2005.
6. John Black, Phillip Rogaway, and Thomas Shrimpton. Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV. In Moti Yung, editor, *CRYPTO*, volume 2442 of *LNCS*, pages 320–335. Springer, 2002.
7. John Black, Phillip Rogaway, Thomas Shrimpton, and Martijn Stam. An Analysis of the Blockcipher-Based Hash Functions from PGV. *J. Cryptology*, 23(4):519–545, 2010.
8. Ewan Fleischmann, Christian Forler, and Stefan Lucks. McOE: A Family of Almost Foolproof On-Line Authenticated Encryption Schemes. In Anne Canteau, editor, *FSE*, volume 7549 of *LNCS*. Springer, 2012.
9. Ewan Fleischmann, Christian Forler, Stefan Lucks, and Jakob Wenzel. McOE: A Family of Almost Foolproof On-Line Authenticated Encryption Schemes (extended version). Cryptology ePrint Archive, Report 2011/644, 2011.
10. Martin E. Hellman. A cryptanalytic time-memory trade-off. *IEEE Transactions on Information Theory*, 26(4):401–406, 1980.
11. Shoichi Hirose. Secure Block Ciphers Are Not Sufficient for One-Way Hash Functions in the Preneel-Govaerts-Vandewalle Model. In Kaisa Nyberg and Howard M. Heys, editors, *Selected Areas in Cryptography*, volume 2595 of *LNCS*, pages 339–352. Springer, 2002.
12. Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable Block Ciphers. *J. Cryptology*, 24(3):588–613, 2011.
13. Bart Preneel, René Govaerts, and Joos Vandewalle. Hash Functions Based on Block Ciphers: A Synthetic Approach. In Douglas R. Stinson, editor, *CRYPTO*, volume 773 of *LNCS*, pages 368–378. Springer, 1993.
14. Jean-Jacques Quisquater and Jean-Paul Delescaille. How Easy is Collision Search. New Results and Applications to DES. In Gilles Brassard, editor, *CRYPTO*, volume 435 of *LNCS*, pages 408–413. Springer, 1989.
15. Phillip Rogaway and Thomas Shrimpton. Deterministic Authenticated-Encryption: A Provable-Security Treatment of the Key-Wrap Problem. Cryptology ePrint Archive, Report 2006/221, 2006.
16. Phillip Rogaway and Haibin Zhang. Online Ciphers from Tweakable Blockciphers. In Aggelos Kiayias, editor, *CT-RSA*, volume 6558 of *LNCS*, pages 237–249. Springer, 2011.