Analysis of the Initial and Modified Versions of the Candidate 3GPP Integrity Algorithm 128-EIA3

Thomas Fuhr, Henri Gilbert, Jean-René Reinhard, and Marion Videau*

ANSSI, France
{thomas.fuhr,henri.gilbert,jean-rene.reinhard,
marion.videau}@ssi.gouv.fr

Abstract. In this paper we investigate the security of the two most recent versions of the message authentication code 128-EIA3, which is considered for adoption as a third integrity algorithm in the emerging 3GPP standard LTE. We first present an efficient existential forgery attack against the June 2010 version of the algorithm. This attack allows, given any message and the associated MAC value under an unknown integrity key and an initial vector, to predict the MAC value of a related message under the same key and the same initial vector with a success probability 1/2. We then briefly analyse the tweaked version of the algorithm that was introduced in January 2011 to circumvent this attack. We give some evidence that while this new version offers a provable resistance against similar forgery attacks under the assumption that (key, IV) pairs are never reused by any legitimate sender or receiver, some of its design features limit its resilience against IV reuse.

Keywords: cryptanalysis, message authentication codes, existential forgery attacks, universal hashing.

1 Introduction

A set of two cryptographic algorithms is currently considered for inclusion in the emerging mobile communications standard LTE of the 3rd Generation Partnership Project 3GPP. It consists of an encryption algorithm named 128-EEA3 and an integrity algorithm named 128-EIA3[1] — that are both derived from a core stream cipher named ZUC. The algorithms ZUC, 128-EEA3, and 128-EIA3 were designed by the Data Assurance and Communication Security Research Center (DACAS) of the Chinese Academy of Sciences.

An initial version of the specifications of 128-EEA3/EIA3 and ZUC, that is referred to in the sequel as v1.4, was produced in June 2010 and published on the

* also with Université Henri Poincaré-Nancy 1 / LORIA, France.

[1] EEA stands for "EPS Encryption Algorithm" and EIA stands for "EPS Integrity Algorithm". EPS (Evolved Packet System) is an evolution of the third generation system UMTS that consists of new radio access system named LTE (Long Term Evolution) and a new core network named SAE (System Architecture Evolution).

A. Miri and S. Vaudenay (Eds.): SAC 2011, LNCS 7118, pp. 230–242, 2012.

GSMA web site for an initial public evaluation [5,6]. Following the discovery of some cryptographic weaknesses in the ZUC v1.4 initialisation [20,18] and of the forgery attack on 128-EIA3 v1.4 reported in this paper, tweaks to the specifications of ZUC and EIA3 were introduced by the designers and a modified version of the specifications referred to in the sequel as v1.5 was published in January 2011 for a second public evaluation period [8,9]. After its adoption by 3GPP, 128-EEA3/EIA3 will represent the third LTE encryption and integrity algorithm set, in addition to the already adopted sets 128-EEA1/EIA1 [4] based on the stream cipher SNOW 3G and 128-EEA2/EIA2 [1, Annex B] based on AES.

The integrity algorithm 128-EIA3 is an IV-dependent MAC that takes as input (1) a 128-bit key, (2) various public parameters that together determine a 128-bit initial vector, (3) an input message of length between 1 and 20000 bits, and produces a 32-bit MAC value. It uses an universal hash function-based construction and has therefore many features in common with the algorithms of the well known Wegman-Carter family of message authentication codes [3,19].

As already mentioned, we denote by 128-EIA3 v1.4 (resp. 128-EIA3 v1.5) the initial version specified in [5] (resp.the modified version specified in [8]). In this paper we analyse the security of both versions. We first show that 128-EIA3 v1.4 is vulnerable to a simple existential forgery attack. Given any known message M, any known or unknown initial vector, and the associated MAC under an unknown key, it is possible to predict the MAC value associated with a new message $M' \neq M$ derived from M under the same initial vector and the same unknown key, with a success probability $1/2$. This attack is generic, it does not rely on any specific feature of ZUC and works with any underlying stream cipher. It exploits a subtle deviation of 128-EIA3 v1.4 from the requirements of the Wegman-Carter paradigm. The latter requirements can be informally summarized by saying that mask values must behave as one-time masks, which is not the case for 128-EIA3 v1.4. As will be shown in the sequel, distinct 128-EIA3 v1.4 mask values are not necessarily independent. Indeed, in 128-EIA3 v1.4, the mechanism used to generate the masking values applied to the output of the universal hash function does not match the model used in the proof. Consequently, the arguments from [12] and [16] that are invoked in the design and evaluation report [7] to infer bounds on the success probability of forgery attacks on 128-EIA3 v1.4 are not applicable.

In [8], a tweak leading to 128-EIA3 v1.5 has been proposed to circumvent this attack. Through an improved generation procedure, masking values are either equal or independent. However, it can be observed that for distinct messages, no separation between the ZUC keystream bits involved in the universal hash function computation and those involved in the generation of the masking values is ensured.

While this represents a deviation from the requirements on masking values used in the Wegman-Carter paradigm, the security consequences are much less dramatic than for the initial MAC (v1.4) since an ad hoc proof given in [10] allows to show that the modified MAC offers a provable resistance against existential forgery attacks under the assumption that the same (key, IV) pair can never

be re-used, neither by the MAC issuer nor by the MAC verifier. We show that this property however affects the resilience of 128-EIA3 v1.5 against forgery attacks if IV repetitions occur. We further observe that independently of this property, the universal hash function structure also results in some limitations of this resilience. This leads us to investigate the resistance of 128-EIA3 v1.5 and one natural variant of this MAC against forgery attacks involving three pairwise distinct messages and the same IV value. We make no claims regarding the practical applicability of the identified nonce repetition attacks to the LTE system.

In Section 3, we give a short description of the 128-EIA3 algorithms. We then describe the attack on v1.4 in Section 4 and discuss the reasons why the security proofs for related constructions by Krawczyk [12] and Shoup [16] do not guarantee the security of 128-EIA3 v1.4. In Section 5, we state a property which, although it may not be considered as an attack in standard security models, underscores the lack of robustness of 128-EIA3 v1.5 against nonce repetition. We also explain why a simple modification of 128-EIA3 fails to completely suppress such properties because of the universal hashing underlying structure.

2 Notation

Throughout the paper, we use the following notation.

- $\mathcal{S}$ is a stream cipher.
- For two finite bitstrings $A = (a^0, \ldots, a^{\ell-1})$ and $B = (b^0, \ldots, b^{m-1})$, $A \| B$ denotes the concatenation of A and B, i.e. the bitstring $(a^0, \ldots, a^{\ell-1}, b^0, \ldots, b^{m-1})$.
- For a bitstring $A = (a^0, \ldots)$ of length $\geq j+1$, $A|_j^i, 0 \leq i \leq j$, denotes the $(j-i+1)$-bit string obtained from the consecutive bits of A between indices i and j, i.e. $A|_j^i = (a^i, \ldots, a^j)$.
- 0^ℓ denotes the bitstring of length ℓ whose bits are all zero.
- $W^{(i)}$ denotes the i-th bit of a 32-bit word W.
- Let consider a 32-bit word $W = (W^{(0)}, \ldots, W^{(31)})$ and an integer a between 1 and 31. Then $W \ll a$ denotes the $(32-a)$-bit word resulting from a left shift of W by a positions and a truncation of the a rightmost bits. More precisely, $W \ll a = (W^{(a)}, \ldots, W^{(31)})$. The $(32-b)$-bit word, $W \gg b$, resulting from the right shift of W by b positions and a truncation of the b leftmost bits is defined in the same way. We have $W \gg b = (W^{(0)}, \ldots, W^{(31-b)})$.[2]

3 Description of the 128-EIA3 Integrity Algorithms

The integrity algorithms 128-EIA3 make a black box use of a stream cipher to generate a keystream from a key and an initial value. A stream cipher $\mathcal{S}$

[2] We are thus using the same somewhat unusual convention as in [6] for defining the symbols "$\gg$" and "$\ll$" as a "shift and truncate" rather than mere shifts. This is motivated by the fact that this convention is more convenient for presenting the attack of Section 4.

is an algorithm that takes as input a k-bit key IK and an n-bit initialisation value IV and outputs a binary sequence $z_0, \ldots, z_i, \ldots$ named the keystream. The keystream is used to compute a 32-bit MAC value Tag according to the procedure described in Algorithm 1 which includes versions v1.4 and v1.5 for conciseness.

Stated differently, the MAC value T associated with IK, IV, and an ℓ-bit message $M = (m_0, \ldots, m_{\ell-1})$ is derived by accumulating (for a set of positions i determined by the message bits and the message length) 32-bit words $W_i = (z_i, \ldots, z_{i+31})$ extracted from the keystream by applying it a 32-bit "sliding window":

$$T = (\bigoplus_{i=0}^{\ell-1} m_i W_i) \oplus W_\ell \oplus W_{mask},$$

where $W_{mask} = W_{L-32}$ with the value L being different between v1.4 and v1.5, i.e. $W_{mask} = W_{\ell+32}$ for v1.4 and $W_{mask} = W_{\lceil \frac{\ell}{32} \rceil \times 32+32}$ for v1.5. The parameter lengths used in 128-EIA3 are: $k = n = 128$ and $1 \leq \ell \leq 20000$.

In fact, the MAC of a message M is computed as

$$MAC(M) = H_{(z_0,\ldots,z_{\ell+31})}(M) \oplus W_{mask},$$

Algorithm 1. The 128-EIA3 MAC algorithms

Input: $IK \in \{0,1\}^k$, $IV \in \{0,1\}^n$, $1 \leq \ell \leq 20000$
Input: $M = (m_0, \ldots, m_{\ell-1}) \in \{0,1\}^\ell$
  if v1.4 **then**
    $L = \ell + 64$
  else if v1.5 **then**
    $L = \left\lceil \frac{\ell}{32} \right\rceil \times 32 + 64$ {**This is the only difference between v1.4 and v1.5**}
  end if
  $(z_0, \ldots, z_{L-1}) \leftarrow \mathcal{S}(IK, IV)|_{L-1}^0$
  $\mathsf{Tag} = 0$
  for $i = 0$ to $\ell - 1$ **do**
    $W_i \leftarrow (z_i, \ldots, z_{i+31})$
    if $m_i = 1$ **then**
      $\mathsf{Tag} \leftarrow \mathsf{Tag} \oplus W_i$
    end if
  end for
  $W_\ell \leftarrow (z_\ell, \ldots, z_{\ell+31})$
  $\mathsf{Tag} \leftarrow \mathsf{Tag} \oplus W_\ell$
  $W_{mask} \leftarrow (z_{L-32}, \ldots, z_{L-1})$
  $\mathsf{Tag} \leftarrow \mathsf{Tag} \oplus W_{mask}$
  return Tag

where $\left(H_{(.)}\right)$ is a family of universal hash functions based on Toeplitz matrices with pseudorandom coefficients taken from a stream cipher output. We have:

$$H_{(z_0,\ldots,z_{\ell+31})}(m_0,\ldots,m_{\ell-1}) = [m_0, m_1, \ldots, m_{\ell-1}, 1] \cdot \begin{bmatrix} z_0 & z_1 & \ldots & z_{31} \\ z_1 & z_2 & \ldots & z_{32} \\ z_2 & z_3 & \ldots & z_{33} \\ \vdots & \vdots & \ddots & \vdots \\ z_\ell & z_{\ell+1} & \ldots & z_{\ell+31} \end{bmatrix}.$$

4 An Existential Forgery Attack against 128-EIA3 v1.4

In this section we describe an attack on the first version of 128-EIA3, that we call 128-EIA3 v1.4. This algorithm has some specific properties that we will now exploit to transform a valid MAC for a message M into a valid MAC for a message M' related to M.

4.1 Description of the Substitution Attack

We can notice that the words W_i derived from the keystream and corresponding to message bits m_i are not independent from each other. More precisely, we have: $W_{i+1} = ((W_i \ll 1), z_{i+32})$.

Moreover the "one-time masks" W_{mask} associated with identical values of IV but different message lengths are related. We have:

$$W_{mask} = (z_{\ell(M)+32}, \ldots, z_{\ell(M)+63}),$$

where $\ell(M)$ denotes the length of the message M. Let us suppose that W_{mask} is the one-time mask generated for the input (IK, IV, M) and W'_{mask} is the one-time mask generated for the input (IK, IV, M'). If $\ell(M') - \ell(M) = \Delta\ell$ with $0 < \Delta\ell < 32$, we have:

$$W'_{mask} = (W_{mask} \ll \Delta\ell, \beta_0, \ldots, \beta_{\Delta\ell-1}),$$

for some bit values β_i. We can use these relations in a substitution attack.

Let us suppose that the adversary knows a valid MAC value T for a given message $M = (m_0, \ldots, m_{\ell-1})$ of length ℓ bits under a given IV value IV and a key IK. This MAC can be transformed with probability $1/2$ into a valid MAC, T', for the $(\ell+1)$-bit message $M' = (0, m_0, \ldots, m_{\ell-1})$ under the same IV value IV and the same key IK.

Let us analyse what happens during the computation of the MAC for M' (under the same IV value IV and the same key IK). The generated keystream $z_0, \ldots, z_{\ell+64}$ is the same as the keystream that was used to compute T, with one extra bit: $z_{\ell+64}$. As a consequence, the words $W_i, 0 \leq i \leq \ell$ are identical. The one-time mask used is $W'_{mask} = (z_{\ell+33}, \ldots, z_{\ell+64}) = ((W_{mask} \ll 1), z_{\ell+64})$. Then, the MAC value T' is given by the following formula:

$$
\begin{aligned}
T' &= \left(\bigoplus_{i=0}^{\ell} m'_i W_i\right) \oplus W_{\ell+1} \oplus W'_{mask} \\
&= \left(\bigoplus_{i=0}^{\ell-1} m_i W_{i+1}\right) \oplus W_{\ell+1} \oplus W'_{mask} \\
&= \left(\bigoplus_{i=0}^{\ell-1} m_i \left((W_i \ll 1), z_{i+32}\right)\right) \oplus (W_\ell \ll 1, z_{\ell+32}) \oplus ((W_{mask} \ll 1), z_{\ell+64}) \\
&= \left(\left(\left(\bigoplus_{i=0}^{\ell-1} m_i W_i\right) \oplus W_\ell \oplus W_{mask}\right) \ll 1, \beta\right) \\
&= (T \ll 1, \beta)\,, \text{ with } \beta = \bigoplus_{i=0}^{\ell-1} m_i z_{i+32} \oplus z_{\ell+32} \oplus z_{\ell+64}.
\end{aligned}
$$

The value $(T \ll 1, \beta)$ is thus a valid MAC for M'. Knowing T, the adversary only needs to guess the value of bit β, which happens with probability $1/2$. This attack can naturally be generalized by recurrence to generate a valid MAC for $(0^r||M)$, with probability 2^{-r}, when $r < 32$: the corresponding tag is then $T_r = ((T \ll r), \beta_0, \ldots, \beta_{r-1})$ for some value of the bits $(\beta_0, \ldots, \beta_{r-1})$.

Equivalently, we have that $T = (\alpha_0, \ldots, \alpha_{r-1}, T_r \gg r)$. This equation enables an adversary to transform a valid MAC IV, T_r for $(0^r||M)$ into a valid MAC for M with probability 2^{-r}.

The attack was checked for $r = 1$ and larger values of r on a few examples, using the implementation programs provided in the annexes of the specification documents [5,6].

4.2 Partial Flaw in 128-EIA3 v1.4 Security Arguments

The Design and Evaluation Report [7] that accompanied version 1.4 erroneously invokes the security proofs of [16] to infer that in the case of 128-EIA3 v1.4, no forgery of a new message can succeed with probability higher than 2^{-32}. The argument comes from the fact that the algorithm makes use of an ε-almost XOR universal (ε-AXU) family of hash functions with $\varepsilon = 2^{-32}$.

Definition 1. *[3,19,17,12,14] A family of hash functions $\{H_K\}_{K \in \{0,1\}^k}$ of range $\{0,1\}^t$ is ε-AXU if for any two distinct messages M, M' in $\{0,1\}^*$ and any $c \in \{0,1\}^t$*

$$Pr_{K \in \{0,1\}^k}[H_K(M) \oplus H_K(M') = c] \leq \varepsilon.$$

In [7], a proof is given that for any value of IV, the family of hash functions used in 128-EIA3, i.e. the intermediate value obtained in the MAC computation associated with key K just before before the exclusive or with W_{mask} is ε-AXU with $\varepsilon = 2^{-32}$.

As far as we know, the first construction of a secure MAC using ε-AXU hash functions has been issued by Krawczyk [12], who proved that given $H_K(M) \oplus r$

for secret uniformly drawn values of K and r, an adversary cannot determine $H_K(M') \oplus r$ with probability higher than ε. The one-time mask generation issue is briefly addressed by noticing that in most practical applications, the mask generation will rely on a stream cipher.

In [14, Appendix B], the security notions related to a Wegman-Carter MAC scheme using a pseudorandom function producing the one-time mask from a counter cnt is stated. In [15, Proposition 14], the probability of a forgery success is computed. The scheme is defined by: a finite PRF $F : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^t$, a counter $\mathsf{cnt} \in \{0,1\}^n$, and a family of universal hash functions $\{H_K\}_{K \in \{0,1\}^k}$. The computation and the verification of MACs require to share an integrity key that consists of a random $a \in \{0,1\}^\kappa$ and a random $K \in \{0,1\}^k$. At most 2^n messages may be MACed with the same key a, and

$$MAC(M) = (\mathsf{cnt}, F_a(\mathsf{cnt}) \oplus H_K(M)).$$

All the models used for the proofs assume that the hash function and the pseudorandom function are randomly chosen and in particular that they are *independent* from each other. In the case of 128-EIA3 v1.4, the construction does not fit the model as the two are related. Moreover, what makes our attack work is that the one-time masks used for messages M and M' of distinct lengths are different but related. In fact, we have:

$$MAC(M) = (\mathsf{cnt}, \mathcal{S}(IK, \mathsf{cnt})|_{\ell(M)+63}^{\ell(M)+32} \oplus H_{\mathcal{S}(IK,\mathsf{cnt})|_{\ell(M)+31}^{0}}(M)).$$

We see that the mask computation also involves the message length and leads to distinct, but related mask values, for identical IVs and different message lengths. Therefore no existing proof applies and we manage to derive an attack against v1.4.

5 Sensitivity of 128-EIA3 v1.5 to Nonce Reuse

In order to resist to our forgery attack, 128-EIA3 has been tweaked [8], leading to the specification of 128-EIA3 v1.5. This new version corresponds to the condition v1.5 in Algorithm 1. The tweak ensures that mask values generated by the algorithm for a given (key, IV) pair for different messages are either equal or independent through an improved selection of the location in the keystream from which the mask value is extracted:

$$W_{mask} = (z_{L-32}, \ldots, z_{L-1}), \text{ with } L = \left\lceil \frac{\ell(M)}{32} \right\rceil \times 32 + 64.$$

This comes at the cost of using a slightly longer part of the keystream. Although this ensures resistance against forgery attacks under the assumptions that (1) neither the MAC issuer nor the MAC verifier reuse any IV value under the same key and the (2) the keystream bits generated by ZUC are indistiguishable from random, as proven in [10, Section 11], we remark that this scheme remains fragile towards IV reuse.

In [15,16], the question of a stateful MAC (implying a counter) against a stateless MAC (with a randomly chosen IV) is briefly discussed. It is underlined in [16] that reliably maintaining a state may be difficult. Practical experience shows that the correct handling of IVs is not a trivial task. Indeed, there is far from a theoretical security requirement to a practical implementation of a scheme and former IV critical modes like CBC have already been subjected to attacks against practical implementations (see e.g. [13]). Therefore we think that it is also important to assess the level of robustness of a scheme in the case of an improper handling of the IV.

In this section we expose two specific properties of 128-EIA3 v1.5, which do not affect a generic Wegman-Carter authentication scheme. These properties involve the MACs of three distinct messages under the same key/IV pair. Therefore, they might threaten the security of 128-EIA3 v1.5 if an adversary can get the MAC of two distinct messages under the same (key, IV) pair. Such an event can happen if IVs are mistakenly repeated by the MAC generating party. It can also happen without deviating from the expected behaviour of the message authentication through substitution attacks: the attacker may use verification queries to gain knowledge on the system [2,11]. More in detail, two valid 128-EIA3 tag values can be obtained by an adversary for the same (key, IV) pair and two distinct messages with a non-negligible probability due to the short MAC size (32 bits): one from the MAC generating party and (with probability 2^{-32}) an extra one from the verifying party. This may allow the adversary to predict with certainty the MAC value of a third message with the same (key, IV) pair.[3]

5.1 On the Independance of Universal Hashing Keys and Masking Values

In the following we consider tags generated using the same key/IV pair. We remark that in the case of 128-EIA3 v1.5, even though masking values for two distinct messages are either equal or independent, the independence of the universal hash function keys (i.e. the keystream bits used in the computation of the hash value) and the masking values is not guaranteed. Parts of the keystream (z_i) used as masking values for a message can be used during the universal hash function computation for a longer message, and conversely. This represents a deviation of the mask value generation of 128-EIA3 v1.5 from the Wegman-Carter paradigm. We show that consequently, while the proof of [10] guarantees that the MACs associated with two distinct messages and the same IV value are independent and uniformly distributed, the knowledge of the tags of two related messages under the same (key, IV) pair may allow to compute the tag of a third message under the same key and IV. Consider any message M_1 of arbitrary length ℓ_1, any message M_2 of length $\ell_2 \geq 1 + 32(\lceil \frac{\ell_1}{32} \rceil + 1)$, and the message $M_3 = M_2 \oplus \delta$ of length $\ell_3 = \ell_2$, where δ is the bitstring of length ℓ_2 whose prefix of length ℓ_1

[3] Whether this third message and the associated tag can be successfully submitted to the verifying entity depends on wheter the IV repetition detection of this entity is effective or not.

is M_1 and whose other bits are zero except for the two bits at positions ℓ_1 and $32(\lceil \frac{\ell_1}{32} \rceil + 1)$. Then we have $MAC(M_1) \oplus MAC(M_2) \oplus MAC(M_3) = 0$. Indeed,

$$\begin{aligned}
MAC(M_1) &= \oplus_{i=0}^{\ell_1 - 1}(m_i^1 W_i) \oplus W_{\ell_1} \oplus W_{32(\lceil \frac{\ell_1}{32} \rceil + 1)}, \\
MAC(M_2) &= \oplus_{i=0}^{\ell_2 - 1}(m_i^2 W_i) \oplus W_{\ell_2} \oplus W_{32(\lceil \frac{\ell_2}{32} \rceil + 1)}, \\
MAC(M_3) &= \oplus_{i=0}^{\ell_1 - 1}(m_i^1 W_i) \oplus \oplus_{i=0}^{\ell_2 - 1}(m_i^2 W_i) \oplus W_{\ell_1} \oplus W_{32(\lceil \frac{\ell_1}{32} \rceil + 1)} \oplus W_{\ell_2} \oplus W_{32(\lceil \frac{\ell_2}{32} \rceil + 1)}.
\end{aligned}$$

Consequently, for any such triplet of pairwise distinct messages the authentication codes of two messages gives a forgery for the third one.

The above 3-message forgery can be avoided by making the masking values and the universal hashing keys independent, for example by following the slightly modified MAC described in Algorithm 2.

Algorithm 2. A modified version of 128-EIA3

Input: $IK \in \{0,1\}^k$, $IV \in \{0,1\}^n$, $\ell \in \mathbb{N}^*$
Input: $M = (m_0, \ldots, m_{\ell-1}) \in \{0,1\}^\ell$
  $(z_0, \ldots, z_{\ell+63}) \leftarrow \mathcal{S}(IK, IV)|_{\ell+63}^0$
  $\mathsf{Tag} = 0$
  $W_{mask} \leftarrow (z_0, \ldots, z_{31})$
  for $i = 0$ to $\ell - 1$ **do**
    $W_i \leftarrow (z_{i+32}, \ldots, z_{i+63})$
    if $m_i = 1$ **then**
      $\mathsf{Tag} \leftarrow \mathsf{Tag} \oplus W_i$
    end if
  end for
  $W_\ell \leftarrow (z_{\ell+32}, \ldots, z_{\ell+63})$
  $\mathsf{Tag} \leftarrow \mathsf{Tag} \oplus W_\ell$
  $\mathsf{Tag} \leftarrow \mathsf{Tag} \oplus W_{mask}$
  return Tag

This algorithm is quite similar to 128-EIA3 and requires the same number of keystream bits and the same amount of computation as 128-EIA3 v1.4 — the single difference being that the mask value consists of the first keystream bits and the universal hash function output value is derived from the subsequent keystream bits. This scheme ensures the equality or independence of keystream bits used as masking values or universal hashing key when tagging two different messages. It is also closer to the Wegman-Carter paradigm in that the masking value computation does not depend on the message being tagged — which is not the case in 128-EIA3 v1.4 and v1.5, where the length of the tagged message impacts the masking value. Unfortunately some non-generic properties remain, that are related to the Toeplitz matrix structure underlying the universal hash function construction rather than to the masking values generation method and hold for both 128-EIA3 v1.5 and Algorithm 2.

5.2 On the Sliding Property of the Universal Hash Function of 128-EIA3

In Section 4 we exploited a sliding property of the universal hash function used by 128-EIA3. Let z be the keystream sequence used in the computation of the universal hash function (i.e. without the final encrypting mask value). We denote by H_z the universal hash function. Using the "sliding-window" property of the construction based on Toeplitz matrices, we can derive the following property. For $r < 32$, we have

$$H_z(0^r \| M) \gg r = H_z(M) \ll r.$$

Let us now consider two messages M and $M' = 0\|M$ and assume that we got their tags T and T' under the same key/IV pair. Assume furthermore that these tag computations involve the same masking value W_{mask}. This is always the case in Algorithm 2 and is true in 128-EIA3 v1.5 under some mild assumption on the length of M (namely that $\ell \pmod{32} \neq 0$). Thus we get

$$\begin{aligned} H_z(M') \oplus W_{mask} &= T', \\ H_z(M) \oplus H_z(M') &= T \oplus T'. \end{aligned}$$

Let us now consider $M'' = 0^2\|M$. We have

$$\begin{aligned} (H_z(M') \oplus H_z(M'')) \gg 1 &= (H_z(0\|M) \oplus H_z(0\|M')) \gg 1 \\ &= (H_z(0\|M) \gg 1) \oplus (H_z(0\|M') \gg 1) \\ &= (H_z(M) \ll 1) \oplus (H_z(M') \ll 1) \\ &= (H_z(M) \oplus H_z(M'))) \ll 1 \\ &= (T \oplus T') \ll 1. \end{aligned}$$

By guessing a single bit, we thus get the value of $H_z(M') \oplus H_z(M'')$. Provided that the computation of the tag of M'' involves the same masking value W_{mask} (i.e. $\ell \pmod{32} \neq 31$ in the case of 128-EIA3 v1.5), by adding $H_z(M') \oplus H_z(M'')$ to T' we get a tag value for M''.

In other words, one can find a triplet (M, M', M'') of pairwise distinct messages such that given the tags T and T' of the first two messages under the same IV, the tag T'' of the third one under the same IV can be guessed with a probability as large as $1/2$. This results from the lack of 2-independence of the universal hash function H_z used in 128-EIA3. While H_z is uniformly distributed and 2^{-32}-AXU — this implies the independence of the MACs of any two distinct messages under the same key and the same IV as shown in [10] — H_z is far from being 2-universal, i.e. the hashes of two distinct messages can be strongly correlated and this results in the lack of independence of the MACs of three pairwise distinct messages illustrated here.[4]

[4] While another choice of H_z might have led to a much lower maximum success probability for a 3-message forgery, the existence of 4-message forgeries of success probability 1 seems difficult to avoid for any $GF(2)$-linear universal function family.

5.3 The IV Construction in 128-EIA-3 and Prevention of Nonce Reuse

The input to the IV construction for 128-EIAx are [1]:

- a 32-bit counter COUNT,
- a 5-bit bearer identity BEARER,
- a 1-bit direction of transmission DIRECTION.

This differs notably from the UMTS Integrity Algorithm (UIA) where the inputs for the IV construction are [4]:

- a 32-bit counter COUNT-I,
- a 32-bit random value FRESH,
- a 1-bit direction of transmission DIRECTION.

In the case of 128-EIA3, the IV is 128 bits and defined by 4 32-bit words, $IV_0 \| IV_1 \| IV_2 \| IV_3$ where:

$$\begin{aligned} IV_0 &= \mathsf{COUNT} \\ IV_1 &= \mathsf{BEARER} \| 0^{27} \\ IV_2 &= IV_0 \oplus \left(\mathsf{DIRECTION} \| 0^{31}\right) \\ IV_3 &= IV_1 \oplus \left(0^{16} \| \mathsf{DIRECTION} \| 0^{15}\right) \end{aligned}$$

We notice that while in UMTS two distinct values managed by the sending and receiving parties ensure the non-repetition of IVs, one single 32-bit counter is used for this purpose in LTE. Enforcing the use of fresh IVs by both the MAC issuer and the MAC verifier might therefore be more complex and we may express some concerns about the assurance that in LTE implementations the strong security requirement of (key, IV) pair never being reused at either side will always be verified.

6 Conclusion

The existential forgery attack presented in Section 4 was forwarded to the designers of 128-EIA3 v1.4, who produced the modified version 128-EIA3 v1.5 to address the issue. While our analysis of 128-EIA3 v1.5 did not reveal any security issue of similar significance and the new MAC offers a provable resistance (under some assumptions) against a large class of forgery attacks, we have highlighted some structural properties of the mask values computation and the universal family of hash functions underlying 128-EIA3 v1.5, and shown that these may lead to limitations of its resilience against nonce reuse. None of the security properties we have investigated here relates to the specific features of the underlying IV-dependent stream cipher ZUC.

Acknowledgements. The authors would like to thank Steve Babbage for insightful comments on an early version of this paper.

References

1. 3GPP Technical Specification Group Services and System Aspects: 3GPP System Architecture Evolution (SAE); Security architecture (Release 9). Tech. Rep. 3G TS 33.401 V 9.3.1, 3rd Generation Partnership Project (2010-04)
2. Bellare, M., Goldreich, O., Mityagin, A.: The Power of Verification Queries in Message Authentication and Authenticated Encryption. Tech. Rep. 2004/309, Cryptology ePrint Archive (2004)
3. Carter, J., Wegman, M.: Universal Classes of Hash Functions. Journal of Computer and System Science 18, 143–154 (1979)
4. ETSI/SAGE: Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2. Document 1: UEA2 and UIA2 Specification. Version 2.1. Tech. rep., ETSI (March 16, 2009), http://www.gsmworld.com/documents/uea2_uia2_d1_v2_1.pdf
5. ETSI/SAGE: Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 1: 128-EEA3 and 128-EIA3 Specification. Version 1.4. Tech. rep., ETSI (July 30, 2010)
6. ETSI/SAGE: Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 2: ZUC Specification. Version 1.4. Tech. rep., ETSI (July 30, 2010)
7. ETSI/SAGE: Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 4: Design and Evaluation Report. Version 1.1. Tech. rep., ETSI (August 11, 2010)
8. ETSI/SAGE: Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 1: 128-EEA3 and 128-EIA3 Specification. Version 1.5. Tech. rep., ETSI (January 4, 2011), http://www.gsmworld.com/documents/EEA3_EIA3_specification_v1_5.pdf
9. ETSI/SAGE: Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 2: ZUC Specification. Version 1.5. Tech. rep., ETSI (January 4, 2011), http://www.gsmworld.com/documents/EEA3_EIA3_ZUC_v1_5.pdf
10. ETSI/SAGE: Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 4: Design and Evaluation Report. Version 1.3, Tech. rep., ETSI (January 18, 2011), http://www.gsmworld.com/documents/EEA3_EIA3_Design_Evaluation_v1_3.pdf
11. Handschuh, H., Preneel, B.: Key-Recovery Attacks on Universal Hash Function Based MAC Algorithms. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 144–161. Springer, Heidelberg (2008)
12. Krawczyk, H.: LFSR-Based Hashing and Authentication. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 129–139. Springer, Heidelberg (1994)
13. Martin Albrecht, K.P., Watson, G.: Plaintext Recovery Attacks Against SSH. In: Proceedings of IEEE Symposium on Security and Privacy 2009, pp. 16–26. IEEE Computer Society (2009)
14. Rogaway, P.: Bucket Hashing and Its Application to Fast Message Authentication. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 29–42. Springer, Heidelberg (1995)
15. Rogaway, P.: Bucket Hashing and its Application to Fast Message Authentication. Journal of Cryptology 12(2), 91–115 (1999)
16. Shoup, V.: On Fast and Provably Secure Message Authentication Based on Universal Hashing. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 313–328. Springer, Heidelberg (1996)

17. Stinson, D.: Universal Hashing and Authentication Codes. Design, Codes and Cryptography 4, 369–380 (1994)
18. Sun, B., Tang, X., Li, C.: Preliminary Cryptanalysis Results of ZUC. Presented at the First International Workshop on ZUC Algorithm, vol. 12 (2010)
19. Wegman, M., Carter, J.: New Hash Functions and Their Use in Authentication and Set Equality. Journal of Computer and System Science 22, 265–279 (1981)
20. Wu, H.: Cryptanalysis of the Stream Cipher ZUC in the 3GPP Confidentiality & Integrity Algorithms 128-EEA3 & 128-EIA3. Presented at the ASIACRYPT 2010 rump session (2010), http://www.spms.ntu.edu.sg/Asiacrypt2010/Rump%20Session-%207%20Dec%202010/wu_rump_zuc.pdf