Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/070631

The New LTE Cryptographic Algorithms EEA3 and EIA3

Verification, Implementation and Analytical Evaluation

Ghizlane Orhanou[*] *and Said El-Hajji*

Département Mathématiques et Informatique, Laboratoire Mathématiques, Informatique et Applications, Université Mohammed V - Agdal, Faculté des Sciences - Rabat, Morocco

Received: 22 Mar. 2013, Revised: 22 Jul. 2013, Accepted: 25 Jul. 2013
Published online: 1 Nov. 2013

Abstract: A new set of cryptographic algorithms is being proposed for inclusion in the ”4G” mobile standard called LTE (Long Term Evolution), and the algorithms are open for public evaluation. The new set of confidentiality and integrity algorithms EEA3/EIA3 is based on the new stream cipher ZUC. In the present paper, we are interested in making the verification and the implementation as well as performing the analytical analysis of the two algorithms, as we have done, in previous works, with the first two sets of the LTE cryptographic algorithms EEA1/EIA1 and EEA2/EIA2. The verification of the confidentiality algorithm results on a correction of the proposed 3GPP algorithm code to meet the specifications requirements.

Keywords: LTE, ZUC, Confidentiality, Integrity, Analytical Evaluation, Implementation

1 Introduction

Security is an important feature of the 3^{rd} and 4^{th} generation of the 3GPP family and its evolution is an important issue. EPS (Evolved Packet System) (or LTE-SAE, Long Term Evolution - Service Architecture Evolution) provides security features in a similar way as its predecessors UMTS (Universal Mobile Telecommunication System) and GSM (Global System Mobile). In addition to the mutual authentication functionality of the network and the user, two other security functions are provided to ensure data security during its transmission over the air interface and through the LTE-SAE system: ciphering of both user and signaling data (in the RRC (Radio Resource Control) layer) in order to ensure their confidentiality, and signaling data integrity protection. For the NAS (Non-Access Stratum) signaling data, both ciphering and integrity are provided [6, 7].
In this context, for the LTE network, two standardized algorithms are required for the radio interface, namely:

–EEA: EPS Encryption Algorithm
–EIA: EPS Integrity Algorithm

Two confidentiality and integrity algorithm sets had already been developed and standardized for LTE. The first set, 128-EEA1 and 128-EIA1, is based on the stream cipher SNOW 3G, and was inherited from the UMTS network. The second set, 128-EEA2 and 128-EIA2, is based on the block cipher AES (Advanced Encryption Standard).

3GPP SA3 agreed in May 2009 on a requirement for a third encryption and integrity algorithm set, 128-EEA3 and 128-EIA3, designed in China, so that the Chinese authorities would permit its use in that country. The resulting algorithm set is based on a core stream cipher algorithm named ZUC, after Zu Chongzhi, the famous Chinese scientist from history.

128-EEA3 is the LTE encryption algorithm defined straightforwardly using ZUC while 128-EIA3 is the LTE integrity algorithm, designed as a Universal Hash Function using ZUC as its core.

The main aim of the present paper is to make the analytical evaluation, the verification and the implementation of this new set of cryptographic algorithms. This study will help us to made a clear idea of this set of algorithms in comparison of those already in use in the UMTS network. So, we will carry out the analytical evaluation of both algorithms EEA3 and EIA3, and then perform the verification and the implementation of the algorithms codes given by the 3GPP documents.

[*] Corresponding author e-mail: ghizlane.orhanou@gmail.com

2 Analytical Evaluation & Verification of the Confidentiality Algorithm EEA3

The confidentiality algorithm 128-EEA3 is a stream cipher that is used to encrypt/decrypt blocks of data under a confidentiality key CK. The maximum length of a block of data may be between 1 and 20000 bits long.
In the following subsections, we study first the EEA3 algorithm operation. Then, we perform an analytical study of the algorithm by calculating its time and space complexity. Finally, after making the verification of the 3GPP algorithm code version, some corrections are done to meet the 3GPP requirements. The result of the implementation of EEA3 algorithm is then presented.

2.1 Confidentiality algorithm EEA3 structure

Figure 1 below illustrates EEA3 operation to encrypt and decrypt messages.

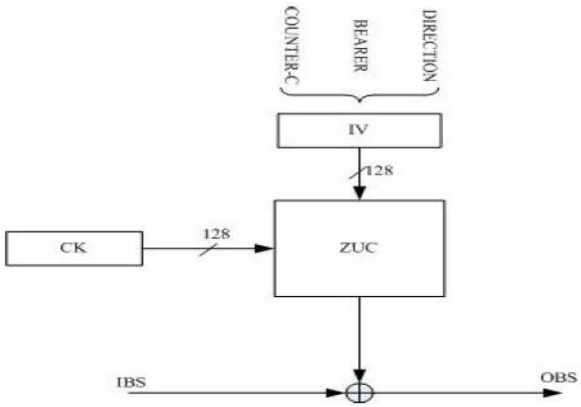

Fig. 1: Principles of the 128-EEA3 encryption operation

Like the other cryptographic algorithms used to ensure data confidentiality in the LTE network (EEA1 and EEA2) [4], the algorithm EEA3 is initialized with the input parameters COUNT-C, BEARER, DIRECTION and the confidentiality key CK. EEA3 uses the stream cipher ZUC as a kernel.
We can distinguish three principal steps during the EEA3 operation: initialization of the keystream generator, keystream generation and finally data encryption/decryption [1].

Before generating the keystream, the keystream generator ZUC is initialized with the input parameters.
Let CK=CK[0] || CK[1] || CK[2] ||, ...,|| CK[15] be the 128-bit confidentiality key, where CK[i](0 $\leq$ i $\leq$ 15) are bytes. We set the 128-bit initial key Key to ZUC as Key = Key[0] || Key[1]|| Key[2] || ... || Key[15], where Key[i](0 $\leq$ i $\leq$ 15) are bytes.
Let COUNT=COUNT[0] || COUNT[1] || COUNT[2] || COUNT[3] be the 32-bit counter, where COUNT[i](0$\leq$ i $\leq$ 3) are bytes. We set the 128-bit initialization vector as IV = IV[0] || IV[1] || IV[2] || ... || IV[15].
ZUC is, then, run to generate key words z[1], z[2], ..., z[L], where
L = $\lceil$ LENGTH / 32 $\rceil$.

Figure 2 below presents the three steps of EEA3 algorithms.

EEA3 Initialization :
For 0 $\leq$ i $\leq$ 15:
    Key[i] = CK[i];
IV[0] = COUNT[0];
IV[1] = COUNT[1];
IV[2] = COUNT[2];
IV[3] = COUNT[3];
IV[4] = BEARER||DIRECTION||00;
IV[5] = IV[6] = IV[7] = 0,
IV[8] = IV[0];
IV[9] = IV[1];
IV[10] = IV[2];
IV[11] = IV[3];
IV[12] = IV[4];
IV[13] = IV[5];
IV[14] = IV[6];
IV[15] = IV[7];

Keystream Generation:
We set L = $\lceil$LENGTH/32$\rceil$
Initialization_ZUC(Key, IV);
/* Allocate memory space for the generated Keystream KS */
KS = (u32*) malloc(4*L);

GenerateKeystream_ZUC(L, *KS);

Encryption/Decryption:
For 0 $\leq$ i $\leq$ LENGTH-1:
    ciphertext[i] = plaintext[i] $\oplus$ KS[i];

Fig. 2: EEA3 algorithm

2.2 EEA3 analytical evaluation

Now, we will focus on the study of the confidentiality algorithm EEA3 time and space complexity. We will start by a brief analytical study of the kernel algorithm.
ZUC, the word-oriented stream cipher, is the basic building block for both 128-EEA3 and 128-EIA3. It is composed of three components with an internal state of 560 bits initialized by a 128-bit cipher key K and a

128-bit initialization vector IV, and outputs a keystream of 32-bit words. This keystream can be used to encrypt the plaintext. Furthermore, ZUC has two operation modes: Initialization Mode and Keystream Mode (or working mode).
During the working mode, the number of words to produce as output depends on the input parameter n. This input parameter presents the length of the plaintext to cipher (or the ciphertext to decrypt). When given as input parameter to ZUC, the algorithm produces exactly n 32-bit output words which will be added by an exclusive-OR operation to the n 32-bit words of the plaintext (or ciphertext) to obtain the ciphertext (or the plaintext). So based on this description and on the ZUC specifications [2], we can see that ZUC has a linear time complexity and constant space complexity.

On the other hand, according to the EEA3 algorithm operation description illustrated in Figure 2 above, EEA3 is based on ZUC algorithm which has a linear time complexity. Besides this, in the last step which concern the encryption (eventually decryption), the number of exclusive-OR operation execution depends, like ZUC Keystream generation, on the length of the text to encrypt or decrypt. These two issues lead to the fact that the EEA3 time complexity is O(n).

Propriety 1*EEA3 algorithm has a linear time complexity.*

On the other hand, during its operation, EEA3 needs to allocate a memory space whose size is 4*L (L is depending on the parameter LENGTH) for the generated keystream. In addition, ZUC used by EEA3 has a constant space complexity. So, the EEA3 space complexity is then O(n).

Propriety 2*EEA3 algorithm has a linear space complexity.*

2.3 EEA3 verification & implementation

After the close study of the EEA3 algorithm, we will expose now the result of the verification of the EEA3 algorithm code version given by the 3GPP standards. We note that the algorithm is coded in C language. Furthermore, we will present the main rectification made to EEA3 algorithm code version to meet the 3GPP requirements. The result of the implementation of the rectified version of the algorithm will be exposed.
During our verification task, we have used the TestSets given by the 3GPP Implementation document [3]. These TestSets are given by the 3GPP for the algorithms implementors to test the correctness of their implementations in respect to 3GPP requirements.

We have found that for a given input data and for a given key and plaintext (Data taken from the TestSets), the ciphertext generated by the algorithm EEA3 in its 3GPP code version was uncorrect [1]. The control of the ciphertext length according the LENGTH parameter value was omitted.

We have, then, added the following instructions to control the length of the ciphertext produced in accordance with the LENGTH parameter value. This issue is very important since the ciphertext and the plaintext must have the same number of bits, but it was omitted in 3GPP code version. You find below the instruction block added:

$$
\begin{aligned}
&j = (LENGTH/32);\\
&j = LENGTH - j*32;\\
&j = 32 - j;\\
&if(length\%32! = 0)\\
&\{\\
&\quad C[L-1] = C[L-1] >> j;\\
&\quad C[L-1] = C[L-1] << j;\\
&\}
\end{aligned}
$$

where L = ⌈ LENGTH/32 ⌉, and C[L-1] is the last block where the length control has to be performed.
It is important to mention that the correction brought to the EEA3 algorithm has no impact on its security and its robustness since it doesn't modify any thing in the algorithm internal processing. You find below one Testset example for the implementation of EEA3 algorithm.

Input Data:
Key = (hex) 17 3d 14 ba 50 03 73 1d 7a 60 04 94 70 f0 0a 29
Count = (hex) 66035492 Bearer = (hex) f
Direction = 0 Length = 193 bits
Plaintext:
(hex) 6cf65340 735552ab 0c9752fa 6f9025fe 0bd675d9 005875b2 00000000

After carrying out the necessary corrections to 3GPP algorithm code, as presented above to make it able to discard the unnecessary bits from the last 64-bit keystream block, we find then the expected result shown in Figure 3 below.

3 Verification & Analytical Evaluation of the Integrity Algorithm EIA3

The new integrity algorithm 128-EIA3 is a Message Authentication Code (MAC) function that is used to compute the MAC of an input message under the control of an integrity key.
In the following subsections, we will study EEA3 operation, make its analytical evaluation and finally perform the verification and then the implementation of the proposed 3GPP algorithm code.

```
EEA3 ALGORITHM ENCRYPTION

        M[i]   XOR   z[i]   =   C[i]

C[0] = 6cf65340 XOR 98e57966 = f4132a26
C[1] = 735552ab XOR 96c22b8e = e5977925
C[2] = 0c9752fa XOR 5c4c744f = 50db26b5
C[3] = 6f9025fe XOR 5b535ee3 = 34c37b1d
C[4] = 0bd675d9 XOR ccb34c57 = c765398e
C[5] = 005875b2 XOR 1bb28711 = 1beaf2a3
C[6] = 00000000 XOR 961d9640 = 80000000
```

Fig. 3: EEA3 implementation result with the rectified code version

3.1 Integrity algorithm EIA3 structure

The integrity algorithm 128-EIA3 computes a 32-bit MAC (*Message Authentication Code*) of a given input message using an integrity key IK. Like EEA3 algorithm, the EIA3 algorithm uses the stream cipher ZUC as a kernel. Figure 4 illustrates the 128-EEA3 structure.

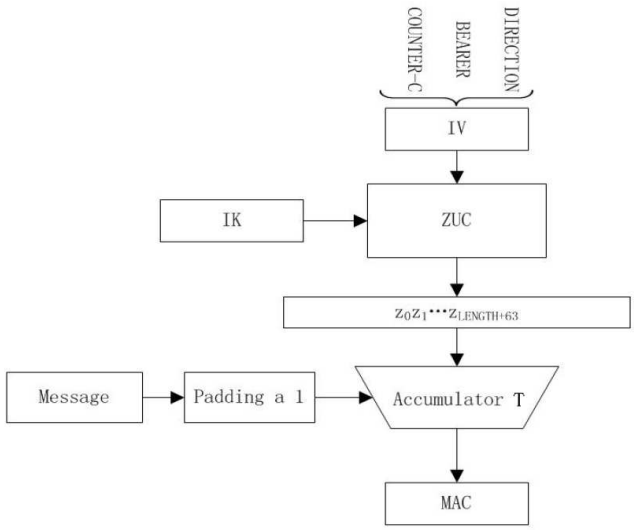

Fig. 4: EIA3 algorithm structure

Prior to processing the message, the ZUC generator is initialized with the integrity key IK and the initialization vector IV, and a keystream with length 64 bits more than the message length is generated. Then, the MAC is calculated. Figure 5 below presents the different steps followed to produce the MAC.

We note N = LENGTH + 64 and L = $\lceil$ N/32 $\rceil$. So, ZUC is running to generate L key words KS[1], KS[2], . . ., KS[L].

We note also k[0], k[1], . . ., k[31], k[32], . . ., k[N-1] the key bit stream corresponding to the previous key words KS.

Furthermore, for each $0 \leq i \leq N-32$, we put k_i = k[i] || k[i+1] || . . . || k[i+31]. So, each k_i is a 32-bit word.

EIA3 Initialization:

For $0 \leq i \leq 15$:
    Key[i] = IK[i];
IV[0] = COUNT[0];
IV[1] = COUNT[1];
IV[2] = COUNT[2];
IV[3] = COUNT[3];
IV[4] = BEARER||000_2;
IV[5] = 00000000_2;
IV[6] = 00000000_2;
IV[7] = 00000000_2;
IV[8] = IV[0] ⊕ (DIRECTION << 7) ;
IV[9] = IV[1] ;
IV[10] = IV[2] ;
IV[11] = IV[3] ;
IV[12] = IV[4] ;
IV[13] = IV[5] ;
IV[14] = IV[6] ⊕ (DIRECTION << 7) ;
IV[15] = IV[7] ;

Keystream Generation:

We set: N = LENGTH + 64;
    L = $\lceil N/32 \rceil$

Initialization_ZUC(Key, IV);

/* Allocate memory space for the generated Keystream KS */
KS = (u32*) malloc(4*L);

GenerateKeystream_ZUC(L, *KS);

MAC Calculation:

T = 0; /* T is a 32-bit word */

For $0 \leq i \leq$ LENGTH - 1:
    if (M[i] = 1), then
        $T = T \oplus k_i$;
$T = T \oplus k_{LENGTH}$;

$MAC = T \oplus k_{N-32}$;

Fig. 5: EIA3 algorithm

3.2 EIA3 analytical evaluation

Figure 5 gives a general overview of the EEA3 operation. During the second step of the EIA3 operation, the generation of the keystream is performed by ZUC algorithm who has a linear time complexity. Furthermore, during the MAC-I calculation step, the T accumulation procedure (see Figure 5) depends on the whole concerned message. So, we can conclude that the time complexity of EIA3 algorithm is O(n).

Propriety 3*EIA3 algorithm has a linear time complexity.*

On the other hand, we can notice that, in the EIA3 algorithm, the keystream generated by ZUC algorithm, whose length depends on the concerned message length, has to be stored for later processing during the MAC-I calculation. So, the amount of temporary storage used during the EIA3 operation depends on the message length. And since ZUC algorithm has a constant space complexity, we can conclude that EIA3 has a linear space complexity.

Propriety 4*EIA3 algorithm has a linear space complexity.*

3.3 EIA3 verification & implementation

As far as the EIA3 algorithm is concerned, the result of our verification shows that the algorithm is coded in respect of all 3GPP requirements, and the result of the implementation tests were as expected by the 3GPP implementation documents. Below, is presented an example of one of the testsets implemented, illustrated in Figure 6.

Input Data:
Key = c9 e6 ce c4 60 7c 72 db 00 0a ef a8 83 85 ab 0a
Count = a94059da Bearer = a Direction = 1
Length = 577 bits
Message:
983b41d4 7d780c9e 1ad11d7e b70391b1 de0b35da 2dc62f83 e7b78d63 06ca0ea0 7e941b7b e91348f9 fcb170e2 217fecd9 7f9f68ad b16e5d7d 21e569d2 80ed775c ebde3f40 93c53881 00000000

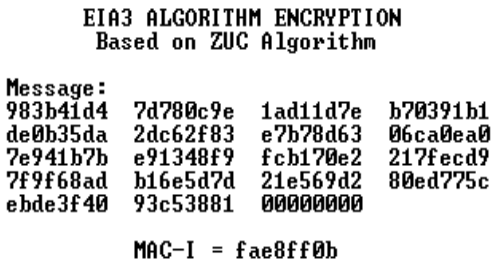

Fig. 6: EIA3 MAC-I calculation

4 Discussion & Conclusion

In this paper, a detailed study of the new LTE confidentiality and integrity algorithms EEA3 and EIA3 based on the stream cipher ZUC has been carried out. The objective is to understand enough EEA3, EIA3 operations and to make an analytical evaluation based on the calculation of their time and space complexities. We have found that the two algorithms have a linear time complexity. Furthermore, both security algorithms EEA3 and EIA3 has linear space complexity.

On the other hand, we have been interested in the verification of EEA3 and EIA3 algorithms, which leads to the rectification of the 3GPP code version of the confidentiality algorithm EEA3 to meet the 3GPP requirements.

At the end of this work we were interested in establishing a comparison, presented in Figure 1 below, of the result of the analytical evaluation of all the cryptographic algorithms used to ensure security features in the Access Network in both UMTS and LTE networks, based on the present paper and our previous works [5].

Table 1: Comparison of the analytical evaluation results of LTE cryptographic algorithms.

Algorithm	Used in	Security goal	Time complexity	Space complexity
UEA2/EEA1	UMTS/LTE	Confidentiality	linear	linear
EEA2	LTE	Confidentiality	linear	constant
EEA3	LTE	Confidentiality	linear	linear
UIA2/EIA1	UMTS/LTE	Integrity	linear	constant
EIA2	LTE	Integrity	linear	linear
EIA3	LTE	Integrity	linear	linear

Thus, it is clear that all the LTE cryptographic algorithms used to ensure either confidentiality or integrity present the advantage of having linear time complexity, which guarantee efficiency and rapidity during the encryption and the decryption process and during the MAC calculation. Furthermore, their space complexity is either linear or constant. These, also, are good results since the maximum length that a message can have is already fixed by 3GPP specifications (20 000 bits) and the Mobile Equipments can easily handle this amount of memory.

References

[1] ETSI/SAGE Specification, Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 1: 128-EEA3 and 128-EIA3 Specification; Version: 1.6, (2011).
[2] ETSI/SAGE Specification, Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 2: ZUC Specification, Version: 1.6, (2011).
[3] ETSI/SAGE Specification, Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 3: Implementor's Test Data, Version: 1.1 (2011).

[4] G. Orhanou, S. El Hajji, Y. Bentaleb and J. Laassiri, EPS Confidentiality and Integrity mechanisms - Algorithmic Approach, International Journal of Computer Science Issues, **7**, 15-23 (2010).
[5] G. Orhanou, S. El Hajji and Y. Bentaleb, EPS AES-based confidentiality and integrity algorithms - Complexity study, Proceeding of the 2^{nd} International Conference on Multimedia Computing and Systems, Ouarzazate, Morocco, (2011).
[6] P. Lescuyer and T. Lucidarme (Both of Alcatel-Lucent, France), Evolved Packet System (EPS) - The LTE and SAE Evolution of 3G UMTS, Jhon Wiley & Sons, Ltd, (2008).
[7] 3GPP TS 33.401, Technical Specification Group Services and System Aspects; 3GPP System Architecture Evolution (SAE): Security architecture; (Release 9), **V9.1.0**, (2009).