Cryptanalysis of the Stream Cipher ZUC in the 3GPP Confidentiality & Integrity Algorithms 128-EEA3 & 128-EIA3

Hongjun Wu, Phuong Ha Nguyen, Huaxiong Wang, San Ling

Nanyang Technological University
Singapore
Stream Cipher ZUC

• 128-EEA3 & 128-EIA3
  – 3GPP confidentiality & integrity algorithms
  – Published in June 2010
  – Currently preliminary version for security evaluation

• The core cryptosystem in 128-EEA3 & 128-EIA3 is stream cipher ZUC
Stream Cipher ZUC

• Based on linear feedback shift register
  – Primitive polynomial over GF($2^{31}$-1)
  – Use addition, XOR, rotation, Sbox to generate keystream
    • high security
Stream Cipher
ZUC: Keystream generation

\[
\mod 2^{31} - 1
\]
Stream Cipher ZUC: Initialization

• Load 128-bit key & 128-bit IV into register
• Run the cipher for 32 steps
  – Each 32-bit keystream is truncated to 31 bits
  – Then XORed to $S_{15}$ (one element of the register)
• Ready for keystream generation
Cryptanalysis of ZUC

• Observation:

If \( a, a', b \in GF(2^{31} - 1) \), \( a \neq a' \),

Then it may happen that

\[
(a \oplus b) \mod(2^{31} - 1) = (a' \oplus b) \mod(2^{31} - 1)
\]
Cryptanalysis of ZUC

• Observation:

Example: \[a = \text{11111111110000000000000000000000}
\]
\[a' = \text{00000000001111111111111111111111}
\]
\[b = \text{11111111110000000000000000000000}
\]

\[(a \oplus b) \mod(2^{31}-1) = (a' \oplus b) \mod(2^{31}-1) = 0\]
Cryptanalysis of ZUC

• Different IVs of ZUC may result in identical keystreams:
  – Introduce difference at iv[0]
  – Difference at $S_{15}$ after feedback
  – Difference at $S_{15}$ may disappear after XORing with the truncated keystream word (based on the previous observation)
    • Then identical states!

Only the first step is involved in the attack!
Cryptanalysis of ZUC

• Probability that identical keystreams appear:
  – For a random key, try all the values of iv[0], iv[10], iv[14], try all the values of the six lsb of iv[15], let iv[2] = 112
    • identical keystreams appear with probability about $2^{-16}$

• Experiment: with difference at iv[0], more than four thousand identical keystream pairs were found
  – after optimizing the search algorithm, finding an identical keystream pairs takes around 3 minutes on a CPU core
Cryptanalysis of ZUC

Examples:

key = 87, 4, 95, 13, 161, 32, 199, 61, 20, 147, 56, 84, 126, 205, 165, 148
iv  = 166, 166, 112, 38, 192, 214, 34, 211, 170, 25, 18, 71, 4, 135, 68, 5
iv’ = 116, 166, 112, 38, 192, 214, 34, 211, 170, 25, 18, 71, 4, 135, 68, 5

keystream words: bfe800d5 0360a22b 6c4554c8 67f00672
2ce94f3f f94d12ba 11c382b3 cbaf4b31 ...

key=79, 104, 119, 45, 239, 93, 93, 202, 172, 113, 158, 37, 85, 121, 134, 148;
iv  =170, 17, 112, 85, 0, 138, 20, 77, 6, 91, 153, 83, 105, 0, 92, 63;
iv’ =128, 17, 112, 85, 0, 138, 20, 77, 6, 91, 153, 83, 105, 0, 92, 63;

keystream words: 0131e501 8f1ef253 6a928250 ded7df1b
fbb9bfe8 e74ce021 1344b122 da9dd837 ...
Cryptanalysis of ZUC

• Key recovery
  – Difference at iv[0] result in identical keystreams
    • The values of key[0] & key[15] are known
    • The sum of key[13]+key[10]*16+ key[4]*8 is known

=> The effective key size is reduced from 128 bits to about 100 bits
Cryptanalysis of ZUC

• Other attacks
  – Difference at iv[1] (estimation)
    • For a random key and an IV pair with difference at iv[1], identical keystream pair appears with probability about $2^{-61}$
      – For $2^8$ IVs with difference at iv [1], identical keystream pair appears with probability $2^{-47}$
  • The effective key size is reduced from 128 bits to about 66 bits
How to resist the attack?

If \(a, a', b \in GF(2^{31} - 1), a \neq a',\)
Then it may happen that
\[(a \oplus b) \mod (2^{31} - 1) = (a' \oplus b) \mod (2^{31} - 1)\]

If \(a, a' \in GF(2^{31} - 1), a \neq a',\)
Then
\[(a + b) \mod (2^{31} - 1) \neq (a' + b) \mod (2^{31} - 1)\]
How to resist the attack?

To resist the attack:

In the initialization stage, “+” instead of “XOR” be used to combine the truncated keystream word with $S_{15}$
Conclusion

• Stream cipher ZUC is weak against chosen IV attack
  – XORing the elements in GF(2^{31}-1) may be non-invertible

• Fixing the security flaw
  – Use only the addition modulo GF(2^{31}-1) when updating the LFSR over GF(2^{31}-1)