SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Single Key Recovery Attacks on Reduced AES-192 and Kalyna-128/256

Gaoli Wang[1,2*] & Chunbo Zhu[1]

[1]*School of Computer Science and Technology, Donghua University, Shanghai* 201620*, China;*
[2]*School of Computer Science and Software Engineering, East China Normal University, Shanghai* 200062*, China*

Appendix A Preliminaries

Appendix A.1 A Brief Description of AES

The Advanced Encryption Standard(AES) is a Substitution-Permutation Network [1]. Three key sizes are available for this iterated block cipher, namely 128, 192 and 256. The 128-bit internal state is treated as a byte matrix of size 4×4, each byte representing a value in $GF(2^8)$ that is defined via the irreducible polynomial $x^8 + x^4 + x^3 + x + 1$ over $GF(2)$. Depending on the key size, N_r rounds are applied to the internal state, e.g., $N_r = 10$ for AES-128, $N_r = 12$ for AES-192 and $N_r = 14$ for AES-256. In each round, there are 4 basic operations:
– SubBytes(SB) applies an 8-bit S-box to each byte of the state in parallel.
– ShiftRows(SR) cyclically rotates the i-th row by i bytes to the left, where i = 0, 1, 2, 3.
– MixColumns(MC) multiplies each column of the state by a constant MDS matrix over $GF(2^8)$.
– AddRoundKey(AK) xors the state with the round subkey.
Note that an additional AddRoundKey operation using a whitening key will be performed before the first round, and the MixColumns operation of the last round is omitted.

The key schedule of AES transforms the master key into N_r + 1 128-bit subkeys. This subkey array can be represented in the form of $W[0, ..., 4 \times N_r + 3]$ where each word $W[\cdot]$ is composed of 32 bits. The length of master key is then denoted by N_k 32-bit words, e.g., $N_k = 4$ for AES-128, $N_k = 6$ for AES-192 and $N_k = 8$ for AES-256. We load the first N_k 32-bit words of $W[\cdot]$ with the master key, and update the rest words of $W[\cdot]$ in the following manner:
– For $i = N_k$ to $4 \times N_r + 3$ do
 • if $i \equiv 0 \bmod N_k$, then $W[i] = W[i - N_k] \oplus SB(W[i - 1] \lll 8) \oplus RCON[i / N_k]$,
 • else if $N_k = 8$ and $i \equiv 4 \bmod 8$, then $W[i] = W[i - 8] \oplus SB(W[i - 1])$,
 • otherwise $W[i] = W[i - 1] \oplus W[i - N_k]$,
where $RCON[\cdot]$ is an array of fixed constants, and $\lll$ denotes circular left rotation. For complete details of AES, we refer to [1].

Appendix A.2 A Brief Description of Kalyna

The Kalyna block cipher [4] was selected as the new Ukrainian encryption standard in 2015. Similar to AES, Kalyna also adopts an SPN structure. In addition, it supports block sizes and key lengths of 128, 256, 512 bits, where the key length can either be equal to or double the block size. Thereby, this block cipher has five variants, namely Kalyna-128/128, Kalyna-128/256, Kalyna-256/256, Kalyna-256/512 and Kalyna-512/512. Of the five variants, we choose Kalyna-128/256 as our target. Hence in the following we only give the description of Kalyna-128/256. For details of other Kalyna variants, the reader is referred to [4].

The internal state for Kalyna-128/256 can be viewed as a byte matrix of size 8×2. After a pre-whitening addition module 2^{64}, an AES-like round function is iterated for 14 times to update the state. To be specific, the round function consists of four transformations:
– SubBytes(SB) applies an 8-bit S-box to each byte of the state in parallel.
– ShiftRows(SR) cyclically rotates the i-th row by $\lfloor \frac{i \cdot b}{512} \rfloor$ bytes to the right, where $0 \leqslant i \leqslant 7$ and b denotes the block size.

* Corresponding author (email: glwang@sei.ecnu.edu.cn)

– MixColumns(MC) multiplies each column of the state by a constant 8×8 MDS matrix over $GF(2^8)$.
– AddRoundKey(AK) xors the state with the round subkey.
Besides, the AK operation of the last round is replaced by a post-whitening addition module 2^{64}.

As regards the key schedule of Kalyna, it is divided into two parts. The first one is the generation of even indexed subkeys, where each even indexed subkey is generated independently from the master key. For the odd indexed subkeys, they can be linearly calculated from the previous round key k_{i-1} according to the formula:

$$k_i = \left(k_{i-1} \lll \left(\frac{l}{4} + 24\right)\right)$$

where l is the length of the block, and $\lll$ denotes circular left rotation.

Such design makes the recovery of the master key from the subkeys infeasible. Therefore, in this article we will not recover the master key, but rather all the round subkeys. For complete description of the key schedule, especially the generation of even indexed subkeys, one may refer to [4].

Appendix A.3 Notations

In the sequel, we will give an account of the notations and definitions utilized in this paper. Moreover, these notations and definitions apply to both AES-192 and Kalyna-128/256.

P and C stand for the plaintext and the ciphertext respectively. Four symbols X_i, Y_i, Z_i, W_i are employed to represent the internal state before SB, SR, MC and AK transformations in the i-th round, where $1 \leqslant i \leqslant N_r$. Besides, the subkey involved in each round is denoted by k_i in accordance to the round number, while the first whitening subkey is denoted by k_0. The 16 bytes of the 128-bit matrix are numbered by column from top to bottom, within the range of 0 to 15. Let $X_i[m]$ denote the state byte in position m in round i, then $X_i[m\text{-}n]$ represents the state bytes positioned from m to n. To refer to the difference in a state X_i, we use the notation ΔX_i. In some cases, we will swap the order of MC and AK operations so as to make the description of the attack procedure more explicit. Since both operations are linear, this modification does not affect the result. Accordingly, we now add the state with an equivalent key $u_i = MC^{-1}(k_i)$ and then perform the transformation MC. The new intermediate state is denoted by $\overline{w}_i$.

In this paper, we measure the memory complexity of the attacks in units of 128-bit AES (or Kalyna) blocks and the time complexity in terms of reduced-round AES (or Kalyna) encryptions.

Property 1 (AES S-box and Kalyna S-box). Given any AES S-box (or Kalyna S-box), say S, and any two non-zero 8-bit differences, say Δ_{in} and Δ_{out}, the equation $S(x) \oplus S(x \oplus \Delta_{in}) = \Delta_{out}$ has one solution on average.

Property 2 (AES Super S-box and Kalyna Super S-box [2]). Given any AES Super S-box (or Kalyna Super S-box) keyed by the subkey k, say SSB_k and any two non-zero 32-bit (or 64-bit) differences Δ_{in} and Δ_{out}, the equation $SSB(x)_k \oplus SSB(x \oplus \Delta_{in})_k = \Delta_{out}$ has one solution on average.

Appendix B The 9-Round Key Recovery Attack on AES-192

Appendix B.1 Proof of Observation 1

Proof. Arguably, the sequence $(e^1_{out} \oplus e^0_{out}, e^2_{out} \oplus e^0_{out}, ..., e^{31}_{out} \oplus e^0_{out})$ is equivalent to the one $(e^1_{in} \oplus e^0_{in}, e^2_{in} \oplus e^0_{in}, ..., e^{31}_{in} \oplus e^0_{in})$. Yet from the path depicted in Figure B1, we discover that $(e^1_{in} \oplus e^0_{in}, e^2_{in} \oplus e^0_{in}, ..., e^{31}_{in} \oplus e^0_{in})$ can be calculated by the following 37 byte parameters:

$$W^i_1[14]||X^i_2[14]||X^i_3[4-7]||X^i_4||k_4[0, 2-5, 7-10, 13-15]||k_5[2, 7, 8] \tag{B1}$$

To prove that, we first denote the difference $W^m_1[14] \oplus W^i_1[14]$ by $\Delta W^m_1[14]$ ($0 \leqslant m \leqslant 31$ and $W^m_1[14] = m$). Then, given the values of $\Delta W^m_1[14]$, $X^i_2[14]$, $X^i_3[4-7]$ and X^i_4, one can easily deduce W^m_4[0,2-5,7-10,13-15]. Afterwards with the knowledge of k_4[0,2-5,7-10,13-15]$||k_5$[2,7,8], it is sufficient to acquire Z^m_6[8,10,11], or more precisely, the value of e^m_{in}. Hence, the target sequence can be obtained by performing this procedure for another 31 times.

However, if the message pair(w^i_1, w^j_1) satisfies the differential characteristic in Figure B1, these 37 byte parameters can be defined by 22 byte variables, namely:

$$\Delta W^j_1[14]||X^i_2[14]||X^i_3[4-7]||Z^i_5[0-11]||Z^i_6[8, 10, 11]||\Delta Z^j_6[8] \tag{B2}$$

Indeed, the knowledge of $\Delta W^j_1[14]||X^i_2[14]||X^i_3[4-7]$ allows us to deduce ΔX^j_4. On the other hand, suppose the values of $Z^i_5[0-11]||Z^i_6[8, 10, 11]||\Delta Z^j_6[8]$ are known, ΔY^j_4 can be calculated backward directly. Then for the fixed difference $\Delta X^j_4||\Delta Y^j_4$, we obtain on average one value of $X^i_4||Y^i_4$ according to Property 1. At the same time, $u_2[6], u_3[1, 4, 11, 14], k_4[0, 2-5, 7-10, 13-15]$ and $k_5[2, 7, 8]$, which are denoted by black spot (•) in Figure B1, are determined, too. Then by the key schedule of AES-192, we have $k_1[14] = SB(k_4[7]) \oplus k_4[14]$. With this subkey, the value of $W^i_1[14]$ is deduced. Consequently, one gets all the 37 byte parameters.

Since there is no key relation between the subkey bytes that are marked by black spot(•) in Figure B1, the key-dependent sieve is of little help in further reducing the size of the sequence. Therefore, we conclude that the sequence $(e^1_{out} \oplus e^0_{out}, e^2_{out} \oplus e^0_{out}, ..., e^{31}_{out} \oplus e^0_{out})$ can assume at most 2^{176} values.

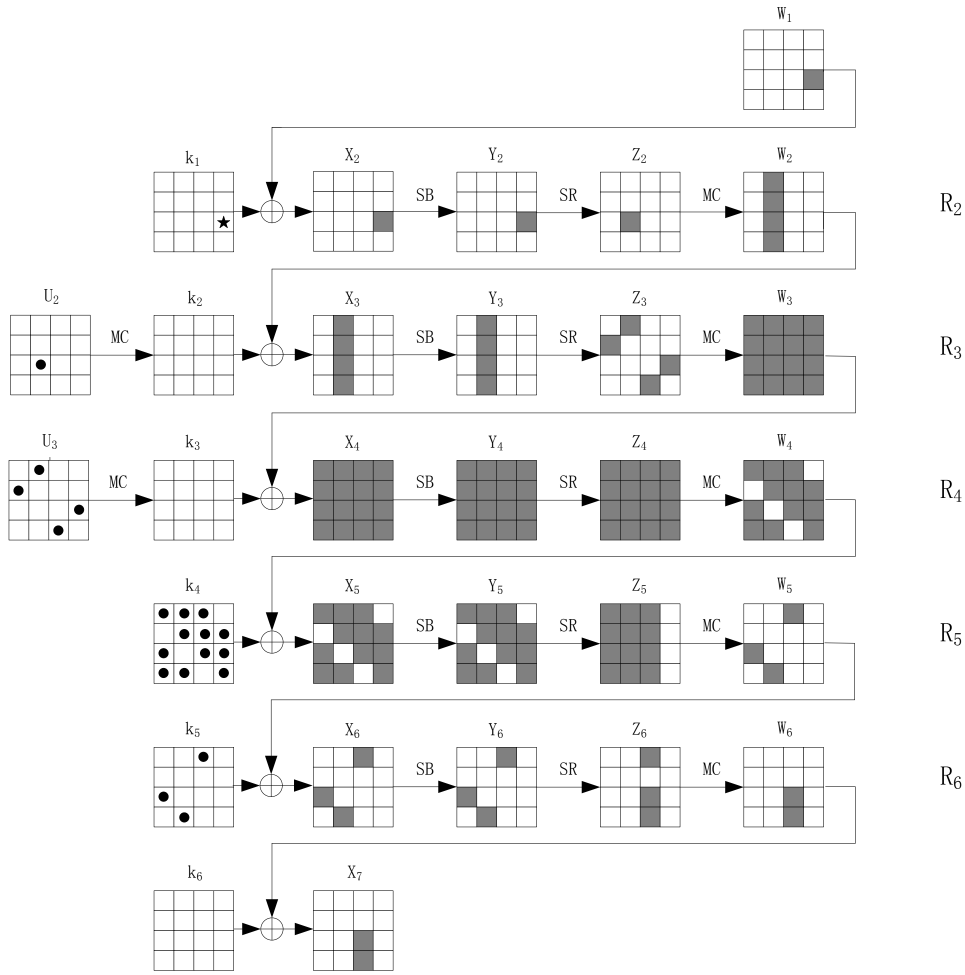

Figure B1 The 5-Round Distinguisher for AES-192(the subkey bytes, marked by black spot (●), can be deduced by the 22 byte variables; while $k_1[14]$, marked by black star ($\star$), can be deduced by the key schedule)

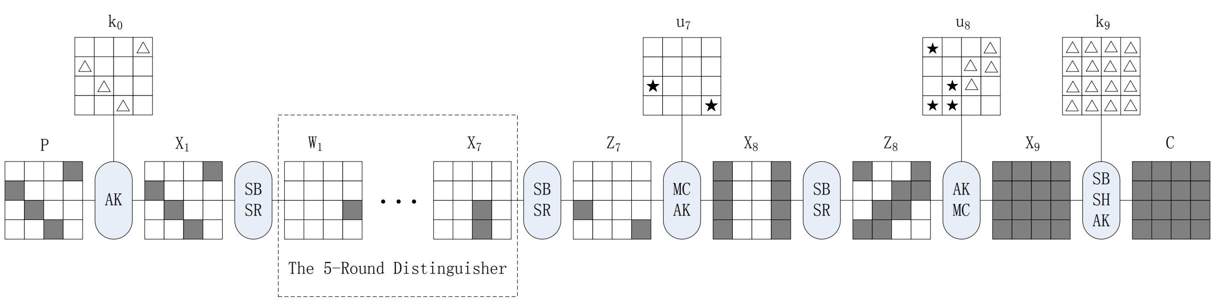

Figure B2 The 9-Round Attack on AES-192(the subkey bytes, marked by triangle (Δ), are the ones we need to guess in the online phase; while the subkey bytes, marked by black star ($\star$), can be deduced by the key schedule)

Appendix B.2 The Attack Procedure

Precomputation Phase. In this phase, two hash tables, named T_1 and T_2, will be built. To begin with, for the table T_1 which contains all the 2^{176} possible sequences, we iterate over the 2^{176} values of the 22 byte variables in (B2). Next, for each of them, evaluate the corresponding 37 byte parameters in (B1). Finally the sequence $(e^1_{out} \oplus e^0_{out}, e^2_{out} \oplus e^0_{out}, ..., e^{31}_{out} \oplus e^0_{out})$ is deduced and stored in the table T_1.

Regarding the table T_2, it is designed to store the values of e_{out}. For all the 2^{80} values of subkeys $u_7[2,15]||u_8[0,3,6,7,9,10,12,13]$, we decrypt $\overline{w_8}[0,3,6,7,9,10,12,13]$ in an attempt to get the corresponding e_{out}. Afterwards, the result is stored with the index of $u_7[2,15]||u_8[0,3,6,7,9,10,12,13]||\overline{w_8}[0,3,6,7,9,10,12,13]$.

Online Phase. This phase is composed of three steps. The first one searches the right pair conforming to the 9-round differential path outlined in Figure B2 by guessing some subkeys. Next, we construct the corresponding δ-set and compute the sequence. Finally the result is matched against the ones in the precomputed table T_1.

1. Encrypt 2^{81} structures of 2^{32} plaintexts where bytes 1,6,11,12 take all the 2^{32} possible values and the rest of the bytes are constants. In total, we can generate 2^{144} pairs among which one is expected to verify the trail shown in Figure B2.
2. For each of the 2^{144} message pairs,
   (a) Choose random differences for the 8 active bytes in Y_8 and propagate them forward to state X_9. Meanwhile, deduce ΔY_9 from the ciphertext difference. Then using Property 1, one value of $X_9||Y_9$ is obtained. Hence, we get k_9. There are as many as 2^{64} suggestions of k_9.

   (b) For each of the 2^{64} suggestions of k_9, deduce $\overline{w_8}[0,3,6,7,9,10,12,13]||u_8[0,3,6,7]$. Then, we compute $\Delta X_8[0,3,14,15]$ and examine whether the result leads to $\Delta Z_7[0,1,3,12-14]=0$. If not, discard the suggestion of k_9. If so, we learn $X_8[1,2,12,13]||Y_8[1,2,12,13]$ from the fixed $\Delta X_8[1,2,12,13]||\Delta Y_8[1,2,12,13]$, on the basis of Property 1. Furthermore, the bytes 9,10,12,13 at u_8 are also known to us. Now we are left with 2^{48} suggestions of $k_9||u_8[0,3,6,7,9,10,12,13]$.

   (c) For each of the 2^{48} suggestions of $k_9||u_8[0,3,6,7,9,10,12,13]$, evaluate $u_7[2]=u_8[6]\oplus u_8[10]$ and $u_7[15]=3\cdot(k_9[0]\oplus k_9[4])\oplus(k_9[1]\oplus k_9[5])\oplus(k_9[2]\oplus k_9[6])\oplus 2\cdot(k_9[3]\oplus k_9[7])$. With these two values and $\Delta X_8[0-3,12-15]||X_8[0-3,12-15]$, the difference at $X_7[10,11]$ could be calculated directly. Then, before moving forward, we need to make sure that $\Delta X_7[10,11]$ result in $\Delta Z_6[9]=0$. In effect, this happens with a possibility of 2^{-8}. Thus, only 2^{40} suggestions are valid for $k_9||u_8[0,3,6,7,9,10,12,13]||u_7[2,15]$.

   (d) Next, deduce $\Delta Y_1[1,6,11,12]$ by guessing $\Delta W_1[14]$. Since ΔX_1, which is consistent with plaintext difference, and ΔY_1 are known, Property 1 enables us to get one value of $X_1[1,6,11,12]$. Arguably, $k_0[1,6,11,12]$ is fixed. Thereby, utmost 2^8 suggestions are possible for $k_0[1,6,11,12]$.

   (e) For each of the 2^8 suggestions of $k_0[1,6,11,12]$, encrypt the message pair through round 1 and retrieve $W_1[12,13,15]$. Let $W_1[14]$ be (0,1,...,31) and compute the corresponding plaintexts($P^0,P^1,...,P^{31}$). Then ask for the encryption of these 32 plaintexts.

   (f) For each of the 2^{40} suggestions of $k_9||u_8[0,3,6,7,9,10,12,13]||u_7[2,15]$, partially decrypt the 32 ciphertexts so as to obtain $\overline{w_8}[0,3,6,7,9,10,12,13]$. Afterwards, look up the table T_2 to get the corresponding e_{out} for each ciphertext by the values of $u_8[0,3,6,7,9,10,12,13]||\overline{w_8}[0,3,6,7,9,10,12,13]||u_7[2,15]$. Then, $(e^1_{out}\oplus e^0_{out},e^2_{out}\oplus e^0_{out},...,e^{31}_{out}\oplus e^0_{out})$ is computed. Discard the subkeys if this sequence is not listed in table T_1.

Recovering the Remaining Subkeys. It should be noted that $k_5[7]=k_9[3]\oplus k_9[7]\oplus k_9[11]\oplus k_9[15]$ by the key schedule. So there are $2^{144}\times 2^{40}\times 2^8/2^8=2^{184}$ subkeys remaining. For each of them, exhaustively search the rest of subkeys.
Attack Complexity. In the precomputation phase, it requires 2^{176} partial encryptions of 32 messages for the construction of table T_1 and 2^{80} partial decryptions of 2^{64} messages for the construction of table T_2. Thus the time complexity of this phase is roughly $2^{176}\times 2^5\times 2^{-0.8}=2^{180.2}$ 9-round AES encryptions. And for the time complexity of online phase, it is apparently dominated by step 2(f), which calls for $2^{144}\times 2^{40}\times 2^8\times 32/2^8=2^{189}$ encryptions if we approximate the complexity of a single AES encryption by 2^8 table lookups as in [3]. Therefore, the overall time complexity is 2^{189} 9-round AES encryptions. Additionally, our attack has a data complexity of 2^{113} chosen plaintexts and a memory requirement of $2^{176}\times 248/128=2^{177}$ 128-bit blocks.

Appendix C The Improved Key Recovery Attack on 9-Round Kalyna-128/256

Appendix C.1 Proof of Observation 2

Proof. Given the 6-round differential path in Figure C1, the multiset $(e^0_{out}\oplus e^i_{out},e^2_{out}\oplus e^i_{out},...,e^{255}_{out}\oplus e^i_{out})$ is actually defined by the following 53 byte parameters:

$$\Delta Z_1^m[7]||X_2^i[0-7]||X_3^i||X_4^i||X_5^i[0-3,12-15]||X_6^i[4-7] \tag{C1}$$

where $\Delta Z_1^m[7]$ stands for the difference $Z_1^m[7]\oplus Z_1^i[7](0\leqslant m\leqslant 255)$.

Yet drilling down, the number of reachable multisets is much less than 2^{424}. This is because if x_1^i belongs to a right pair (x_1^i,x_1^j) that follows the differential trial in Figure C1, these 53 byte parameters depend on the 39 byte variables, which are

$$\Delta Z_1^j[7]||X_2^i[0-7]||X_3^i||Z_5^i[0-7]||Z_6^i[12,13]||\Delta Z_6^j[12-15] \tag{C2}$$

Without any doubt, the adversary can easily get Δx_4^j once he knows the values of $\Delta Z_1^j[7]||X_2^i[0-7]||X_3^i$. From the bottom side, the knowledge of $\Delta Z_6^j[12,13]||Z_6^i[12,13]$ supports us to calculate $\Delta W_5^j[4,5]$. Then, by the MC operation, we are capable of determining $\Delta Z_5^j[0,1,4-7]||\Delta W_5^j[6,7]$ as well as $\Delta X_6^j[6,7]$. Since $\Delta X_6^j[6,7]||\Delta Y_6^j[6,7]$ is known, using Property 1, one value of $X_6^j[6,7]||Y_6^j[6,7]$ is obtained. Then with $\Delta Z_5^j[0,1,4-7]||Z_5^i[0-7]$, it's easy to deduce ΔY_4^j. Afterwards, the adversary finds one value on average for $X_4^i||Y_4^i$ with Property 1. By this means, we get the knowledge of the 53 byte parameters.

In the meantime, the 39 byte variables also fully determine the values of $u_2[0-3,12-15]||k_3||k_4[0-3,12-15]||k_5[4-7]$. Nevertheless, it should be noted that there are some key relations between these subkey bytes. In short, once we know k_3 and $k_5[6-8]$, we get $u_2[0-3,12-15]||k_4[12-14]$ for free. Hence, there are only 2^{224} multisets left after applying the key-dependent sieve which can screen out 2^{88} wrong values.

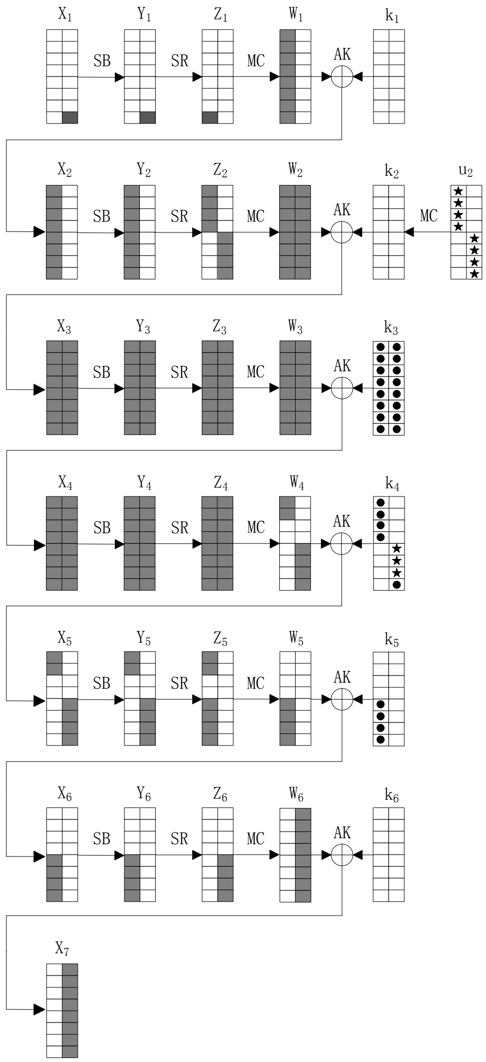

Figure C1 The 6-Round Distinguisher for Kalyna-128/256

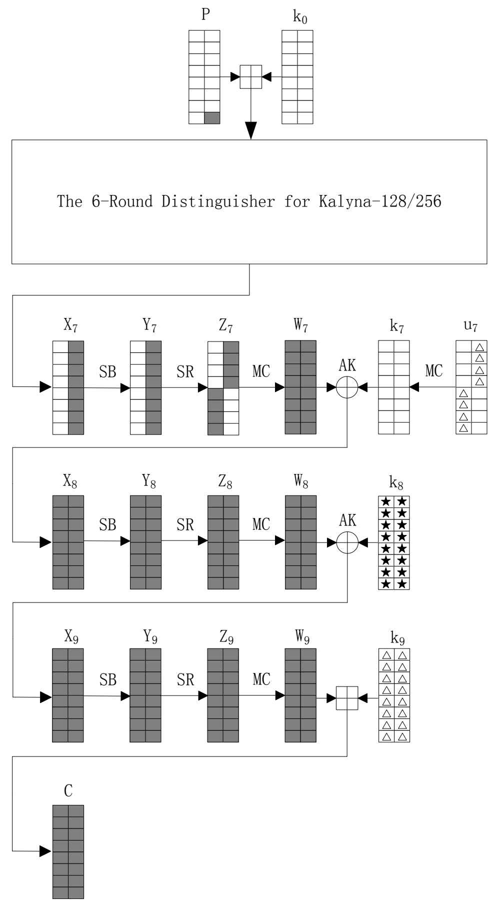

Figure C2 The 9-Round Attack on Kalyna-128/256

Appendix C.2 The Attack Procedure

Precomputation Phase.

1. Iterate over $\Delta Z_6[12-15]||\Delta W_5[4,5]$, and compute $\Delta W_5[6,7]||\Delta Z_5[0,1,4-7]$ by the MC operation. After that, propagate $\Delta W_5[4-7]$ forward to X_6, meanwhile propagate $\Delta Z_6[12-15]$ backward to Y_6. Then it is adequate to get $X_6[4-7]||Y_6[4-7]$. Next, guess $Z_5[0-7]$ to acquire the values of $\Delta W_4[0,1,12-15]||X_5[0-3,12-15]||k_5[4-7]$, by which the bytes 12,13,14 at k_4 are also fixed. Then with the knowledge of $k_4[12-14]||X_5[12-14]$, one learns $W_4[12-14]$. Eventually, we store $X_5[0-3,12-15]||X_6[4-7]$ in a table T_4 with the index of $\Delta W_4[0,1,12-15]||W_4[12-14]$. For each index, there are 2^{40} values of $X_5[0-3,12-15]||X_6[4-7]$.

2. For each $\Delta Z_2[0-3,12-15]||\Delta W_4[0,1,12-15]||k_3$, one deduces $\Delta X_3||\Delta Y_4$. Then, we are able to find one value of $X_3||Y_4$ on average, according to Property 2. After evaluating $u_2[0-3,12-15]$ from k_3 by the key schedule, we learn $Z_2[0-3,12-15]$. Then $X_3||X_4||W_4[0-3,12-15]||\Delta W_4[0,1,12-15]$ are stored in a table T_5 by the index of $\Delta Z_2[0-3,12-15]||Z_2[0-3,12-15]$. On average, each index has 2^{-16} entries.

3. For each $\Delta Z_1[7]||X_2[0-7]$, compute $\Delta Z_2[0-3,12-15]||Z_2[0-3,12-15]$. By this value, the adversary looks up the table T_5 to get $X_3||X_4||W_4[0-3,12-15]||\Delta W_4[0,1,12-15]$. Then for each $\Delta W_4[0,1,12-15]||W_4[12-14]$, 2^{40} values of $X_5[0-3,12-15]||X_6[4-7]$ can be retrieved by accessing the table T_4. Consequently, all the 53 byte parameters are deduced.

4. The last step is to compute the multiset and store it in the table T_3. Furthermore, with the purpose of recovering the remaining subkeys later, we also keep the record of the corresponding $k_3||X_4||X_5[0,1,12-15]||X_6[4-7]$ along with the multiset. From the viewpoint of information theory, we can represent such an entry on $512+336=2^{9.7}$ bits. Hence, the table T_3 requires a storage of $2^{224}\times 2^{9.7}/128=2^{226.7}$ 128-bit blocks.

Online Phase. As shown in Figure C2, the distinguisher is extended by adding 3 more rounds at the bottom. So the effect of the carry bits resulting from the pre-whitening key addition module 2^{64} could be avoided. Unfortunately, on account of the post-whitening key addition module 2^{64}, we have to test all the 2^{128} possible values of k_9. Here are the details:

1. Ask for the encryptions of 2^{97} structures of 2^8 plaintexts where byte 15 assumes all possible values and the remaining bytes are constants.
2. For each of the 2^{112} pairs,
   (a) Traverse the 2^{128} values of k_9 and deduce the corresponding k_8 by the key schedule. We then partially decrypt the ciphertexts through 2 rounds to acquire both the value and difference of X_8. After that deduce $\Delta W_7||\Delta Z_7$ and discard the wrong guesses which don't result in $\Delta Z_7[0-3, 12-15] = 0$. Then only 2^{64} values of $k_9||k_8$ will remain.
   (b) For each of the 2^{64} values of $k_9||k_8$, guess $\Delta Z_6[12-15]$ so as to calculate $\Delta X_7[8-15]$. For $\Delta X_7[8-15]$ is already known, according to Property 1, it is expected that one value of $X_7[8-15]||Y_7[8-15]$ will be found. At the same time, $u_7[4-11]$ is determined. There are at most 2^{32} values of $u_7[4-11]$.
   (c) We now pick one message of the pair, say P_0, to construct the δ-set. This can be done by computing $P_i = P_0 \oplus i$, where $1 \leqslant i \leqslant 255$. Next, query the encryption of the δ-set.
   (d) Using the 2^{96} values of $k_9||k_8||u_7[4-11]$, partially decrypt the ciphertexts and evaluate the corresponding multisets. Then look for a match in the table T_3. If there is no match, the subkeys are eliminated.

Recovering the Remaining Subkeys. Once a match is found in the table T_3, we can affirm with certainty that the guess of $k_9||k_8||u_7[4-11]$ is correct. We now proceed to recover the rest of subkeys. More detailed, for P_0 which is the chosen plaintext in the previous phase, do as follows:

1. Guess the remaining 8 unknown bytes of u_7, and compute the corrsponding k_7 and k_6. With the knowledge of $k_7||k_6$, it is adequate to calculate X_6 from X_8. Then compare $X_6[4-7]$ with the one stored with the matching multiset in the table T_3. If there is no difference between these two values, continue to the next step. Otherwise, discard the guess. It is expected that 2^{32} guesses of $k_7||k_6$ survive.
2. Iterate over k_5 and get the corresponding k_4. Using these two subkeys, the adversary learns $X_4||X_5$ from X_6. He then checks whether $X_4||X_5[0, 1, 12-15]$ is equal to the one obtained from the table T_3 corresponding to the correct multiset sequence. Ultimately, the adversary is left with one value of $k_9||k_8||k_7||k_6||k_5||k_4$.
3. Using k_3 retrieved from the table T_3, compute $k_2||X_2$. Next, traverse k_1, and deduce k_0 as well as the plaintext. Apparently, the calculated plaintext and P_0 must be identical if we find the right subkeys. This happens with a possibility of 2^{-128}. Thereupon, only one value of $k_9||k_8||k_7||k_6||k_5||k_4||k_3||k_2||k_1||k_0$ will remain.

Attack Complexity. We first discuss the time complexity of the precomputation phase. In this phase, a total of 3 tables, including T_3, T_4 and T_5, are established. Concretely, the table T_4 costs $2^{(8+6)\times 8} \times 2^{-2.2} = 2^{109.8}$ encryptions, whereas the table T_5 needs $2^{(16+8+6)\times 8} \times 2^{-2.2} = 2^{237.8}$ encryptions. After that, the table T_3 is constructed at the price of $2^{224} \times 2^8 \times 2^{-0.6} = 2^{231.4}$ encryptions. In the online phase, the time complexity is primarily consumed by the step 2(a) recovering $k_9||k_8$, which is equivalent to $2^{112} \times 2^{128} \times 2^{-2.2} = 2^{237.8}$ encryptions. As for the last phase, we need to perform $2^{32+128} \times 2^{-2.2} = 2^{157.8}$ encryptions for the recovery of k_5. To summarize, the time complexity of our attack is $2^{238.8}$, while the data complexity is 2^{105} chosen plaintexts and the memory requirement for the precomputation table T_3 is $2^{226.7}$ 128-bit blocks.

Appendix D The 10-Round Key Recovery Attack on Kalyna-128/256 from the Second Round

Appendix D.1 Proof of Observation 3

Proof. Let $\Delta Z_2^m[7]$ represent the difference $Z_2^m[7] \oplus Z_2^i[7]$, where $0 \leqslant m \leqslant 255$. Then, we claim that the knowledge of the following 62 byte parameters is sufficient to calculate the multiset $(e_{in}^0 \oplus e_{in}^i, e_{in}^1 \oplus e_{in}^i, ..., e_{in}^{255} \oplus e_{in}^i)$, which is equal to $(e_{out}^0 \oplus e_{out}^i, e_{out}^2 \oplus e_{out}^i, ..., e_{out}^{255} \oplus e_{out}^i)$:

$$\Delta Z_2^m[7]||X_3^i[0-7]||X_4^i||X_5^i||X_6^i||X_7^i[0, 12-15] \tag{D1}$$

However, under the condition that x_2^i belongs to a right pair (x_2^i, x_2^j) which follows the differential trial in Figure D1, then the 62 byte parameters can be deduced by 47 byte variables, namely

$$\Delta Z_2^j[7]||X_3^i[0-7]||X_4^i||Z_6^i||Z_7^i[0, 4-7]||\Delta Z_7^j[0] \tag{D2}$$

For one thing, $\Delta Z_2^j[7]||X_3^i[0-7]||X_4^i$ allows us to compute ΔX_5^j. For another, ΔY_5^j is defined by $Z_6^i||Z_7^i[0, 4-7]||\Delta Z_7^j[0]$. Together, we are expected to find one value of $X_5^i||Y_5^i$ by using Property 1. This also leads to the determination of $k_4||k_5$. By checking whether there is a key relation between these two subkeys, the key-dependent sieve can filter out 2^{128} wrong values of the byte variables. In other words, only $2^{47\times 8-128} = 2^{248}$ values of multiset are valid.

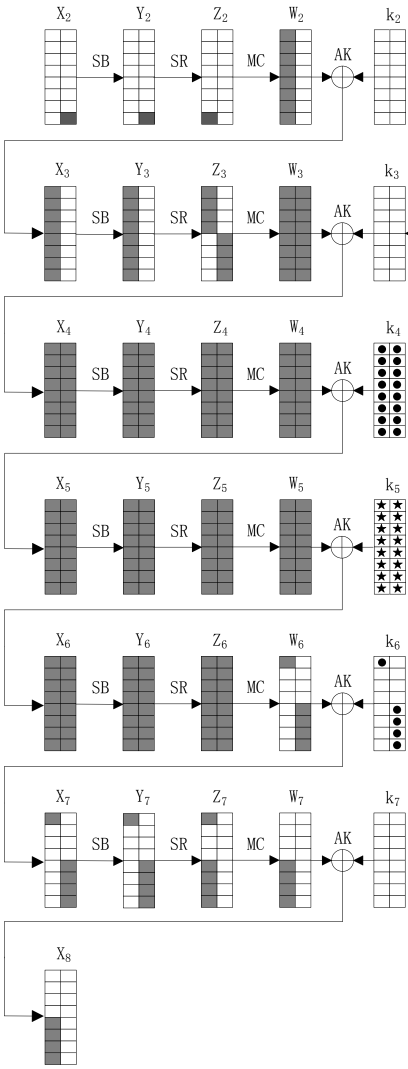

Figure D1 The New 6-Round Distinguisher for Kalyna-128/256

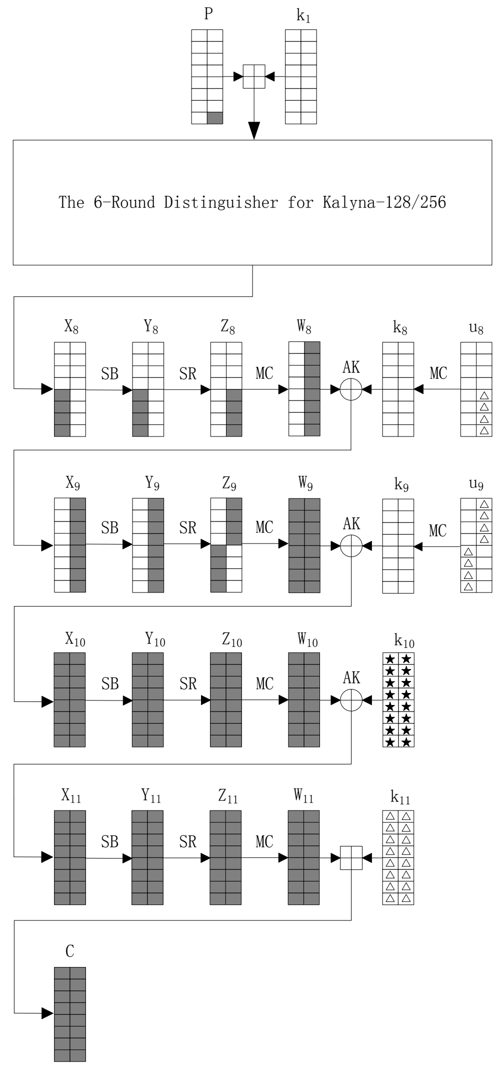

Figure D2 The 10-Round Attack on Kalyna-128/256

Appendix D.2 The Attack Procedure

Precomputation Phase.

1. For each $\Delta Z_7[0]$, deduce $\Delta W_7[[4-7]]||\Delta Z_7[4-7]$ by the MC operation. Then iterate over $\Delta W_6[0,12-15]$, by which $\Delta X_7[0,12-15]$ is fixed. We now adopt Property 1 to get $X_7[0,12-15]$ and $Y_7[0,12-15]$. After that, the table T_7 is built so as to store $X_7[0,12-15]$ with the index of $\Delta W_6[0,12-15]$. It is expected that each index has 2^8 values of $X_7[0,12-15]$.

2. For each $\Delta W_4[0-3,12-15]||k_5[0-7]||\Delta W_6[0,12-15]$, acquire the corresponding $X_5[0-3,12-15]||X_6[0-7]$ by utilizing Property 2. As $k_5[0-7]$ is known, $k_4[7-14]$ is no longer a mystery to us. This enables us to obtain the knowledge of $W_4[12-14]$. Afterwards, $k_4[7-11]||X_5[0-3,12-15]||X_6[0-7]||\Delta W_6[0,12-15]$ is indexed by $\Delta W_4[0-3,12-15]||W_4[12-14]$ and stored in the table T_8. Accordingly, there are 2^{80} entries for each index.

3. Next, we create another table T_9 through the same way as in step 2. In brief, according to Property 2, we find one value of $X_5[4-11]||X_6[8-15]$ for each $\Delta W_4[4-11]||k_5[8-15]||\Delta W_6[0,12-15]$. Using $k_4[0-6,15]$ deduced from $k_5[8-15]$, one can easily get $W_4[4-6]$. Ultimately, we store $k_4[4-6]||X_5[4-6]||X_6[8-15]$ by the index of $\Delta W_4[4-11]||W_4[4-6]||k_4[0-3,15]||X_5[7-11]||\Delta W_6[0,12-15]$ in the table T_9. The average entries for each index is 2^{-40}.

4. Iterate over the $2^{25\times 8}$ values of $\Delta Z_2[7]||X_3[0-7]||X_4$, and evaluate $W_4||\Delta W_4$. By $\Delta W_4[0-3,12-15]||W_4[12-14]$, we look up the table T_8 for the values of $k_4[7-11]||X_5[0-3,12-15]||X_6[0-7]||\Delta W_6[0,12-15]$. From $X_5[0-3,15]$ and $W_4[0-3,15]$, $k_4[0-3,15]$ is deduced. In addition, we learn $X_5[7-11]$ from $k_4[7-11]||W_4[7-11]$. The adversary then is able to retrieve $k_4[4-6]||X_5[4-6]||X_6[8-15]$ from the table T_9 by the index of $\Delta W_4[4-11]||W_4[4-6]||k_4[0-$

$3, 15]||X_5[7-11]||\Delta W_6[0, 12-15]$. Afterwards, for each $\Delta W_6[0, 12-15]$, we access the table T_7 to get 2^8 values of $X_7[0, 12-15]$.

5. For each of the 2^{248} values of the 47 byte parameters, calculate the corresponding multiset. Finally, in order to recover all the subkeys later, we store the multiset with a 45-byte parameter, which is $u_3[0-3, 12-15]||k_5||X_6||X_7[0, 12-15]$, in the table T_6. According to the information theory, each entry can be presented by $512 + 360 = 2^{9.8}$ bits. In that case, the memory complexity of table T_6 is about $2^{248} \times 2^{9.8}/128 = 2^{250.8}$ 128-bit blocks.

Online Phase.

1. In order to find the right pair that satisfies the truncated differential path in Figure D2, we enquire the encryptions of 2^{105} structures of 2^8 plaintexts.
2. For each of the 2^{120} pairs, do:
   (a) Guess k_{11}, in return we get k_{10} for free. With these subkeys, partially decrypt the ciphertexts through 2 rounds to get the difference at Z_9. Then check whether $\Delta Z_9[0-3, 12-15] = 0$. If not, eliminate the guess.
   (b) For each of the remaining 2^{64} values of $k_{11}||k_{10}$, guess $\Delta Z_8[12-15]$ and compute $\Delta X_9[8-15]$. Using Property 1, the adversary obtains $X_9[8-15]||Y_9[8-15]$. Accordingly, $u_9[4-11]$ is determined.
   (c) Next, we learn $\Delta X_8[4-7]$ by guessing $\Delta Z_7[0]$. For $\Delta Y_8[4-7]$ is already known, again Property 1 helps us acquire $X_8[4-7]||Y_8[4-7]$. From $Y_8[4-7]$ and $X_9[8-15]$, we deduce $u_8[12-15]$.
   (d) Take one plaintext of the pair, say P_0, to construct the δ-set. Afterwards encrypt the δ-set.
   (e) For each of the $2^{64+32+8}$ values of $k_{11}||k_{10}||u_9[4-11]||u_8[12-15]$, do partial decryptions over these 256 ciphertexts to state X_8. Then compute the multiset and look up the result in the table T_6. If no match, discard the key guess. Otherwise, we retrieve the corresponding 45-byte parameter, namely $u_3[0-3, 12-15]||k_5||X_6||X_7[0, 12-15]$, and move to the next phase.

Recovering the Remaining Subkeys. At this point, we already know the subkeys $k_{11}||k_{10}||u_9[4-11]||u_8[12-15]||k_5||u_3[0-3, 12-15]$. For P_0, the chosen plaintext, we do the follows:

1. Decrypt the ciphertext to state X_{10} with $k_{11}||k_{10}$. Then guess $u_9[0-3, 12-15]$ and evaluate the corresponding $k_9||k_8||u_8$. As $u_8[12-15]$ is already fixed, each guess can be verified by comparing these two values. Only 2^{32} guesses of $k_9||k_8$ will pass this test.
2. For each of the 2^{32} guesses of $k_9||k_8$, evaluate X_7 by trying all the 2^{128} possible values of k_7. After filtering the wrong guesses which bring about the inconsistence between the deduced $X_7[0, 12-15]$ and the one stored in the table T_6, we are left with 2^{88} values of k_7. Then calculate k_6 as well as X_6. There is a possibility of 2^{-128} that the result matches with the correct X_6. Hence, only one value of $k_9||k_8||k_7||k_6$ is expected to survive.
3. Next, we learn k_4 from k_5. With these two subkeys, one can easily obtain X_4.
4. As $u_3[0-3, 12-15]$ is known, we guess the rest unknown bytes of u_3 and compute $k_3||k_2$. Then for all the 2^{128} values of k_1 and 2^{64} values of $k_3||k_2$, decrypt X_4 through 2 rounds to obtain the plaintext. Compare the result with P_0. This leaves us with 2^{64} values of $k_9||k_8||k_7||k_6||k_5||k_4||k_3||k_2||k_1$.
5. To further screen the remainign subkey guesses, we now pick another plaintext P_1 by computing $P_1 = P_0 \oplus 1$ and ask its encryption. For each of the 2^{64} values of $k_{11}||k_{10}||k_9||k_8||k_7||k_6||k_5||k_4||k_3||k_2||k_1$, decrypt the ciphertext. Only one of the deduced plaintexts is expected to match with P_1. In that case, we claim the right $k_{11}||k_{10}||k_9||k_8||k_7||k_6||k_5||k_4||k_3||k_2||k_1$ is found.

Attack Complexity. In the precomputation phase, obviously the time complexity is determined by the construction of the table T_6, which costs $2^{256-0.7} = 2^{255.3}$ encryptions. When in the online phase, due to the post-whitening key addition module 2^{64} in step 2(a), we have to perform 2^{128} partial decryptions on 2^{120} message pairs. Hence, the time complexity is approximately $2^{120+128-2.3} = 2^{245.7}$ encryptions. Besides, the time complexity of recovering the remaining subkeys is defined by step 4, which requires 2^{192} encryptions. All in all, the whole attack has a time complexity of $2^{255.3}$ encryptions, a data complexity of 2^{113} chonsen plaintexts, and a memory complexity of $2^{250.8}$ 128-bit blocks.

Data/Time/Memory Tradeoff With data/time/memory tradeoff, the adversary can precompute only $2^{248-2} = 2^{246}$ possible values of the multisets in the table T_6. In return, he has to repeat the attack 2^2 times to offset the probability of the failure. Thus, the data complexity increases to 2^{115} chosen plaintexts, while memory requirements decreases to $2^{248-2} \times 2^{9.8}/128 = 2^{248.8}$ 128-bit blocks. Moreover, now the time complexities of the precomputation and the online phases are $2^{255.8-2} = 2^{253.3}$ and $2^{245.7+2} = 2^{247.7}$, respectively. To conclude, our attack can be done with $2^{253.3}$ encryptions.

Acknowledgements This work is supported by the National Natural Science Foundation of China (Nos. 61572125 and 61373142), High Technology Field of "Action Plan for Scientific and Technological Innovation" in Shanghai (No. 16511101400).

References

1 Daemen J, Rijmen V. AES proposal: Rijndael. Journal of Research of the National Institute of Standards and Technology, 1999(1): 97-105

2 Daemen J, Rijmen V. Understanding two-round differentials in AES. In: Ostrovsky R, Prisco R, Visconti I eds. Security and Cryptography for Networks. Lect Notes in Computer Science, Vol 4116. Berlin: Springer-Verlag, 2006. 7894

3 Ferguson N, Kelsey J, Lucks S, et al. Improved cryptanalysis of Rijndael. In: Matsui M, ed. Fast Software Encyption. Lect Notes in Computer Science, Vol 2355. Berlin: Springer-Verlag, 2001. 213230

4 Oliynykov R, Gorbenko I, Kazymyrov O, et al. A new encryption standard of Ukraine: the kalyna block cipher. IACR Cryptol. ePrint Arch. 2015, 650 (2015). http://eprint.iacr.org/2015/650