SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Key-dependent cube attack on reduced Frit permutation in Duplex-AE modes

Lingyue QIN[1], Xiaoyang DONG[1], Keting JIA[2*] & Rui ZONG[1]

[1]*Institute for Advanced Study, Tsinghua University, Beijing* 100084*, China;*
[2]*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084*, China*

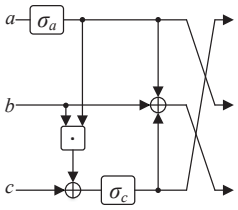
Appendix A

Appendix A.1 Brief description of Frit

Frit is a 384-bit cryptographic permutation proposed by Simon *et al.*, which operates on a state of three limbs a, b, c in $\{0,1\}^{128}$ updated in 16 rounds. The details are illustrated in Algorithm A1.

Algorithm A1 Frit

Input: $a, b, c \in \{0,1\}^{128}$
  for each $i \in [0, 15]$ **do**
    $c \leftarrow c \oplus RC_i$
    $a \leftarrow a \oplus (a \lll 110) \oplus (a \lll 87)$
    $c \leftarrow c \oplus (a \odot b)$
    $c \leftarrow c \oplus (c \lll 118) \oplus (c \lll 88)$
    $b \leftarrow a \oplus b \oplus c$
    $(a, b, c) \leftarrow (c, a, b)$
  end for
  return (a, b, c)



We use Frit to design authenticated encryption by using the duplex authenticated encryption mode as shown in Figure A1. Our attack target is the initialization phase of Frit-AE, as shown in Figure A2.

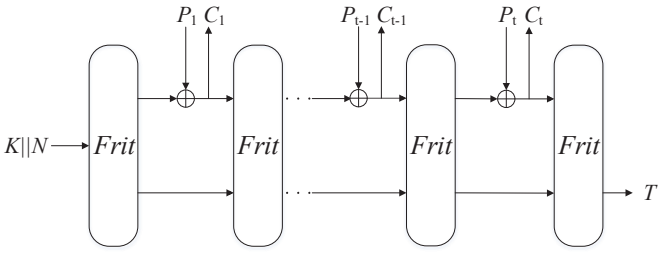


Figure A1 Frit$_\alpha^\beta$-AE.

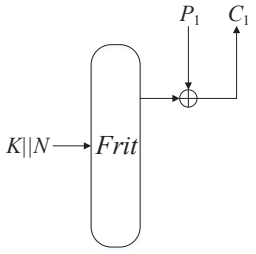


Figure A2 Initialization phase of Frit-AE.

Appendix A.2 Key-dependent cube attack

The key-dependent cube attack only involves conditions which only involve secret key bits. In duplex authenticated encryption mode, such as Ketje, Ascon and Frit$_\alpha^\beta$-AE, the initialization phase produces l-bit output. Each of the output

* Corresponding author (email: ktjia@mail.tsinghua.edu.cn)

bits is written as a polynomial $f_i(k_0,...,k_{n-1},v_0,...,v_{m-1})$, $i=0,1,...,l-1$. Choose a common cube C_T, e.g $(v_0,...,v_{s-1})$, $1 \leqslant s \leqslant m$, then $f_i = T \cdot P_i + Q_i$, $i=0,1,...,l-1$. In our key-dependent cube attack, a common divisor of P_i is found, which is a polynomial $g(k_0,...,k_{n-1})$ that only involved some key bits. Then the cube sum of f_i over all values of the cube C_T is $P_i = g(k_0,...,k_{n-1}) \cdot P_i'$. The Corollary 1 is given.

Corollary 1. Given a series of polynomials f_i $(i \in \{0,1,...,l-1\})$:$\{0,1\}^n \to \{0,1\}$.

$$\begin{cases} f_0(k_0,...,k_{n-1},v_0,...,v_{m-1}) = T \cdot g(k_0,...,k_{n-1}) \cdot P_0' + Q_0 \\ f_1(k_0,...,k_{n-1},v_0,...,v_{m-1}) = T \cdot g(k_0,...,k_{n-1}) \cdot P_1' + Q_1 \\ ... \\ f_{l-1}(k_0,...,k_{n-1},v_0,...,v_{m-1}) = T \cdot g(k_0,...,k_{n-1}) \cdot P_{l-1}' + Q_{l-1} \end{cases} \tag{A1}$$

where none of the monomials in $Q_i(x)$ is divisible by T. Then the sums of f_i $(i \in \{0,1,...,l-1\})$ over all values of the cube (cube sum) are

$$\begin{cases} \sum\limits_{v' \in C_T} f_0(k_0,...,k_{n-1},v',v_s,...,v_{m-1}) = g(k_0,...,k_{n-1}) \cdot P_0' \\ \sum\limits_{v' \in C_T} f_1(k_0,...,k_{n-1},v',v_s,...,v_{m-1}) = g(k_0,...,k_{n-1}) \cdot P_1' \\ ... \\ \sum\limits_{v' \in C_T} f_{l-1}(k_0,...,k_{n-1},v',v_s,...,v_{m-1}) = g(k_0,...,k_{n-1}) \cdot P_{l-1}' \end{cases} \tag{A2}$$

where the C_T contains all binary vectors of the length s, and other public variables $v_j, j \in \{s,s+1,...,m-1\}$ are constants.

The following Property 1 is easy to get. According to Property 1 and Assumption 1, we obtain the cube tester used in our attack.

Property 1. If $g=0$, cube sums of f_i $(i \in \{0,1,...,l-1\})$ will be all 0 with probability 1.

Assumption 1. If $g=1$, cube sums of f_i $(i \in \{0,1,...,l-1\})$ will be determined by P_i' $(i \in \{0,1,...,l-1\})$, the cube sums of f_i $(i \in \{0,1,...,l-1\})$ all equal to 0 with probability about 2^{-l} if f_i $(i \in \{0,1,...,l-1\})$ is a random oracle.

Appendix A.3 Attacks on Frit$_a^b$-AE

We give a 3-round initial structure of Frit$_a^b$-AE by keeping the limb b_0 to constants 0 and setting $\sigma_c^{-1}\sigma_a^{-1}(v)$ to limb c_0 as Figure A3. We generate the constraints set CF of $\{x_i\}$ to guarantee there are no quadratic terms in a_3, c_3 as Algorithm A2.

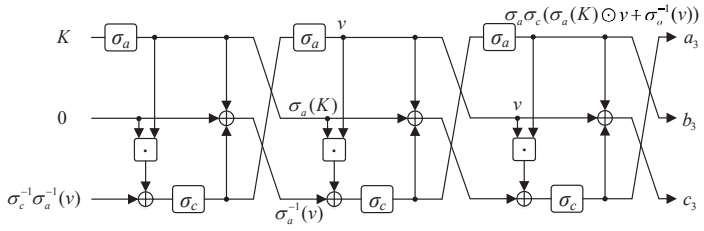


Figure A3 3-round initial structure of Frit$_a^b$-AE.

Algorithm A2 Generating constraints on v to linearize a_3, c_3

Input: Variables set $v = \{v_i\}$ $(i \in [0,127])$
Output: A set CF of constraints

```
CF = ∅
Exp = σ_a σ_c(σ_a(K) ⊙ v) ⊙ v + σ_c(v) ⊙ v
for each i ∈ [0,127] do
    for each j ∈ [i+1,127] do
        if g_{i,j}(K) v_i v_j ∈ Exp and g_{i,j}(K) ≠ 0 then
            CF ← CF ∪ {x_i + x_j ⩽ 1}
        end if
    end for
end for
return CF
```

Then our problem is modeled into a binary linear programming problem:

$$Maximize \sum_{i=0}^{127} x_i,$$
$$s.t.\ AX \leqslant b,\ X = \{x_i | x_i \in \{0,1\}, 0 \leqslant i \leqslant 127\},$$

where the $AX \leqslant b$ describe the constraints set CF. Using the Gurobi Optimizer [1] to solve the problem, we get the first two optimum solutions and the corresponding index sets of v are listed in Table A1. Every variable v_i in each set does not multiply with each other in the same set. In the following we use the $Index_0$ to introduce the basic idea of our attack. (The $Index_0$ can be replaced with $Index_1$ to get different bit conditions of K.)

Table A1 Index sets of independent variables

Set	Num	Values
$Index_0$	29	$0, 1, 7, 8, 15, 16, 23, 30, 31, 38, 39, 45, 46, 53, 60, 61,$ $68, 69, 75, 76, 83, 91, 98, 99, 105, 106, 113, 114, 121$
$Index_1$	28	$0, 1, 2, 9, 16, 23, 24, 25, 31, 32, 39, 46, 54, 55, 61, 62,$ $69, 84, 85, 91, 92, 98, 99, 107, 114, 115, 121, 122$

The procedure to attack $r+3$-round Frit$_a^b$-AE is concluded as follows.

1. First set the cube's dimension $d = F_{r+1} + 1$. Adding $v_j (j \notin Index_0)$ to the cube variables set, we can choose one quadratic term $g_{i,j}(K)v_iv_j (i \in Index_0)$ from c_3 and add v_i to the cube variables set. The other $d-2$ cube variables are choosing from $Index_0$, which are not multiplied with v_j. That is, we obtain a d-dimension cube to recover one bit condition $g_{i,j}(K)$.
2. Assign the other variables of v to constants 0 except for the cube variables and calculate the cube sum of the whole 128 bits output after $r+3$-round Frit. If all the 128 cube sums are 0, we take the $g_{i,j}(K)$ as 0, otherwise $g_{i,j}(K) = 1$.
3. The time complexity of recovering 1 bit condition of K is 2^d. By changing the value of j and relative quadratic term $g_{i,j}(K)v_iv_j$, we can generate different cube variables to recover different $g_{i,j}(K)$. We can get 128 linearly independent bit conditions and solve the set of equations to recover the 128-bit key. We introduce the details to choose different bit conditions and corresponding cube variables in Algorithm A3. The time complexity is $2^d \times 2^7 = 2^{7+d}$. (The time to solving the linear system can be omitted.)

Algorithm A3 Generating bit conditions and corresponding cube variables

Input: A set $Index$, the dimension d
Output: A list B_c of bit conditions and a list C_T of corresponding cube variables

```
B_c = [ ]
C_T = [ ]
Exp = σ_a σ_c(σ_a(K) ⊙ v) ⊙ v + σ_c(v) ⊙ v
for each j ∈ [0, 127] \ Index do
    V_0 = ∅
    V_1 = [ ]
    for each i ∈ Index do
        if g_{i,j}(K)v_iv_j ∈ Exp and g_{i,j}(K) ≠ 0 then
            V_0 ← V_0 ∪ {i}
            if g_{i,j}(K) and (g_{i,j}(K) + 1) not in B_c then
                Add i to V_1
                Add g_{i,j}(K) to B_c
            end if
        end if
    end for
    for each i ∈ V_1 do
        cube = {j, i} ∪ {k_m | k_m ∈ Index \ V_0, 0 ⩽ m ⩽ d − 3}
        Add cube to C_T
    end for
end for
return B_c, C_T
```

Appendix A.3.1 *Experiments on 10-round Frit$_a^b$-AE*

Applying the 3-round initial structure in Figure A3 to the 10-round Frit$_a^b$-AE, we can use the 22-dimension cube to get some bit conditions of K. For example, setting $j = 4$ (v_4 is a cube variable), there are three quadratic terms in the expressions

of c_3 and a_3:

$$(K_4+K_{91}+K_{114})v_4v_{45}, (K_{50}+K_{73}+K_{91})v_4v_{91}, (K_{73}+K_{96}+K_{114})v_4v_{114}.$$

Keeping only one variable in set $\{v_{45}, v_{91}, v_{114}\}$ to be a cube variable, there is only one quadratic term in the expressions of c_3 and a_3. We can get 1 bit condition of K by testing one cube. The examples of the bit conditions and relative cube variables are listed in Table A2. All the 128 bit conditions and corresponding cube variables can be found by Algorithm A3 using SageMath [2]. Then solving a set of 128 linear equations we can recover the 128-bit key. Testing about 100 random keys has a success rate of 100%, and recovering each key needs about 8 minutes with time complexity 2^{29}.

Table A2 Bit conditions and cube variables of 10-round Frit$_a^b$-AE

Bit conditions	Degree	Cube variables
$K_4+K_{91}+K_{114}$	22	$v_0, v_1, v_4, v_7, v_8, v_{15}, v_{16}, v_{23}, v_{30}, v_{31}, v_{38}, v_{39}$ $v_{45}, v_{46}, v_{53}, v_{60}, v_{61}, v_{68}, v_{69}, v_{75}, v_{76}, v_{83}$
$K_{50}+K_{73}+K_{91}$	22	$v_0, v_1, v_4, v_7, v_8, v_{15}, v_{16}, v_{23}, v_{30}, v_{31}, v_{38}, v_{39}$ $v_{46}, v_{53}, v_{60}, v_{61}, v_{68}, v_{69}, v_{75}, v_{76}, v_{83}, v_{91}$
$K_{73}+K_{96}+K_{114}$	22	$v_0, v_1, v_4, v_7, v_8, v_{15}, v_{16}, v_{23}, v_{30}, v_{31}, v_{38}, v_{39}$ $v_{46}, v_{53}, v_{60}, v_{61}, v_{68}, v_{69}, v_{75}, v_{76}, v_{83}, v_{114}$

Appendix A.4 Attacks on Frit$_b^b$-AE

Set variable vector v' to limb a_0 and $\sigma_c^{-1}\sigma_a^{-1}(v)$ to limb c_0 as Figure A4 to get a 2-round initial structure for Frit$_b^b$-AE.

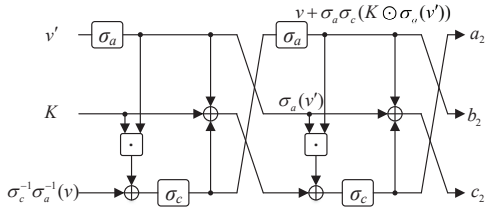


Figure A4 2-round initial structure of Frit$_b^b$-AE.

To linearize a_2 and c_2, we need to keep that the expression $b_1 \odot b_2 = \sigma_a(v') \odot v + \sigma_a(v') \odot \sigma_a\sigma_c(K \odot \sigma_a(v'))$ doesn't have quadratic terms. That is,

1. For expression $\sigma_a(v') \odot v$, we need to keep that each v_i $(0 \leqslant i \leqslant 127)$ is not multiplied by v'_j $(0 \leqslant j \leqslant 127)$ after mixing operation σ_a. So if v'_j is chosen as a cube variable, variables v_j, v_{j+18} and v_{j+41}[1)] need to be constants due to the diffusion property of σ_a.
2. For expression $\sigma_a(v') \odot \sigma_a\sigma_c(K \odot \sigma_a(v'))$, the quadratic term $g_{i,j}(K)v'_iv'_j(i \neq j)$ depends on some relative bits of K. For a certain K, if all $g_{i,j}(K) = 0$, the expression is linear. In the attack procedure, we can set some v'_is to constants to reduce the num of bit conditions $g_{i,j}(K)$.
3. By carefully choosing some variables v_i and v'_j and setting others to constants, we ensure that there are no quadratic terms $v_iv'_j$ in $b_1 \odot b_2$. For all the quadratic terms $g_{i,j}(K)v'_iv'_j$: if $g_{i,j}(K) = 0$ or at least one of v'_i, v'_j is constant, the degree of a_2, c_2 is 1; otherwise the degree is 2.

The key-dependent attack on $r+2$-round Frit$_b^b$-AE is concluded as follows:

1. First set the cube's dimension $d = F_{r+1} + 1$ and cube variables set $C_i = \{v'_i, v'_{i+1}, v_{j_0}, \cdots, v_{j_{d-3}}\}$, where set $\{j_0, \cdots, j_{d-3}\}$ doesn't have any elements of $\{i, i+18, i+41, i+1, i+19, i+42\}$.
2. Assign the other variables of v, v' except for the cube C_i to constants 0 and calculate the cube sums of the whole 128 bit positions of the output limb after $r+2$-round Frit over all values of the cube C_i. If all the 128 cube sums are 0, we take the K_{i+1} as 0, otherwise $K_{i+1} = 1$.
3. The time complexity of recovering 1-bit key is 2^d, and the time to get the whole 128-bit key is $2^d \times 2^7 = 2^{7+d}$ by traversing i from 0 to 127.

1) The addition + is in $GF(2^7)$, i.e. $i+1$ means $(i+1) \mod 128$.

Appendix A.4.1 *Experiments on* 9*-round Frit$_b^b$-AE*

We do experiments on the 9-round Frit$_b^b$-AE to verity our attack results. Using the 2-round initial structure in Figure A4, we can use a (F_8+1)-dimension (22-dimension) cube to recover 1-bit K. The cube variables for recovering K_1 are listed in Table A3. To recover K_i $(0 \leqslant i \leqslant 127)$, the cube variables needed are the variables in Table A3 by adding $i-1$ to the indexes in $GF(2^7)$. We give several examples of the recovered 1-bit key and corresponding 128-bit cube sums for some random keys in Table A4, using the cube variables in Table A3. The details of the experiments refer to https://github.com/qly14/FritAE.git. We test about 100 random keys, and the success rate of recovering the whole 128-bit key is 100%. The time complexity of our attack on 9-round Frit$_b^b$-AE is 2^{29}, which only needs about 7 minutes on a personal computer.

Table A3 Cube variables of 9-round Frit$_b^b$-AE

Key	Degree	Cube variables
K_1	22	$v'_0, v'_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_{10}, v_{11}, v_{12}$
		$v_{13}, v_{14}, v_{15}, v_{16}, v_{17}, v_{20}, v_{21}, v_{22}, v_{23}, v_{24}$

Table A4 Experimental results of 9-round Frit$_b^b$-AE

1-bit key	128-bit random key	Cube sums
$K_1=0$	0x1c93b7ae 81cf5ca8 644a0463 0c41db9e	0x00000000 00000000 00000000 00000000
$K_1=1$	0xe58ec52a 3b3fccf2 17d04d42 4618e031	0x0800c010 20000040 00000000 00802020
$K_2=0$	0x05ab60a7 fe41288e 69983eed 4ae9fe4c	0x00000000 00000000 00000000 00000000
$K_2=1$	0xe96f359e 26ace184 1565c5cb 0fe1b095	0x04006008 10000020 00000000 00401010
$K_3=0$	0x8047f929 e59445dc 0d13ea46 60acb0ec	0x00000000 00000000 00000000 00000000
$K_3=1$	0xb3e808b5 a9094cb4 1064fa84 339eac56	0x02003004 08000010 00000000 00200808

Appendix A.4.2 *Experiments on* 10*-round Frit$_b^b$-AE*

Adding 8-round Frit after the 2-round initial structure, we can attack 10-round Frit$_b^b$-AE using the (F_9+1)-dimension (35-dimension) cube. Similar to the attack on 9-round Frit$_b^b$-AE, we give the cube variables for recovering the K_1 of the 10-round Frit$_b^b$-AE in Table A5. The time complexity is 2^{35} for recovering 1-bit key and 2^{42} for all 128-bit key. Limited to the personal computer power, we only try to recover K_1 for a certain key as an example. The success rate of testing 10 random keys is 100%, and recovering each 1-bit key needs about 8 hours.

Table A5 Cube variables of 10-round Frit$_b^b$-AE to recover K_1

Key	Degree	Cube variables
K_1	35	$v'_0, v'_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}, v_{16}, v_{17}$
		$v_{20}, v_{21}, v_{22}, v_{23}, v_{24}, v_{25}, v_{26}, v_{27}, v_{28}, v_{30}, v_{31}, v_{32}, v_{33}, v_{34}, v_{35}, v_{36}, v_{37}$

Appendix A.4.3 *Attacks on* 11*-round Frit$_b^b$-AE*

Using the 2-round initial structure we can choose the 56-dimension cube to attack the 11-round Frit$_b^b$-AE. The time complexity of recovering 128-bit key is $2^{56} \times 2^7 = 2^{63}$. The cube variables to recover K_1 for 11-round Frit$_b^b$-AE are given in Table A6.

Table A6 Cube variables of 11-round Frit$_b^b$-AE to recover K_1

Key	Degree	Cube variables
K_1	56	$v'_0, v'_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}, v_{16}, v_{17}, v_{20}, v_{21}$
		$v_{22}, v_{23}, v_{24}, v_{25}, v_{26}, v_{27}, v_{28}, v_{29}, v_{30}, v_{31}, v_{32}, v_{33}, v_{34}, v_{35}, v_{36}, v_{37}, v_{38}, v_{39},$
		$v_{43}, v_{44}, v_{45}, v_{46}, v_{47}, v_{48}, v_{49}, v_{50}, v_{51}, v_{52}, v_{53}, v_{54}, v_{56}, v_{57}, v_{59}, v_{60}, v_{61}, v_{62}$

Appendix A.4.4 *Attacks on* 12*-round Frit$_b^b$-AE*

Similar to the previous attack, the 90-dimension cube can be used to attack 12-round Frit$_b^b$-AE with complexity $2^{90} \times 2^7 = 2^{97}$. The cube variables to recover K_1 for 12-round Frit$_b^b$-AE are given in Table A7 as an example.

Table A7 Cube variables of 12-round Frit$_b^b$-AE to recover K_1

Key	Degree	Cube variables
K_1	90	$v_0', v_1', v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}, v_{16}, v_{17}, v_{20}, v_{21}$
		$v_{22}, v_{23}, v_{24}, v_{25}, v_{26}, v_{27}, v_{28}, v_{29}, v_{30}, v_{31}, v_{32}, v_{33}, v_{34}, v_{35}, v_{36}, v_{37}, v_{38}, v_{39},$
		$v_{40}, v_{43}, v_{44}, v_{45}, v_{46}, v_{47}, v_{48}, v_{49}, v_{50}, v_{51}, v_{52}, v_{53}, v_{54}, v_{56}, v_{57}, v_{58}, v_{59}, v_{60}$
		$v_{61}, v_{62}, v_{63}, v_{64}, v_{65}, v_{66}, v_{67}, v_{68}, v_{69}, v_{70}, v_{71}, v_{72}, v_{73}, v_{74}, v_{75}, v_{76}, v_{77}, v_{78}$
		$v_{79}, v_{80}, v_{81}, v_{82}, v_{83}, v_{84}, v_{85}, v_{86}, v_{87}, v_{88}, v_{89}, v_{90}, v_{91}, v_{92}, v_{93}, v_{94}$

Appendix A.5 Attacks on Frit$_c^b$-AE

We give a 4-round initial structure of Frit$_c^b$-AE by keeping the limb a_0 to constants 0 and setting $\sigma_c^{-1}\sigma_a^{-1}(v)$ to limb b_0 as Figure A5. The procedure to attack $r+4$-round Frit$_c^b$-AE is similar with the procedure to attack $r+3$-round Frit$_a^b$-AE. We notice that only the 128 independent equations used to recover the 128-bit key are different.

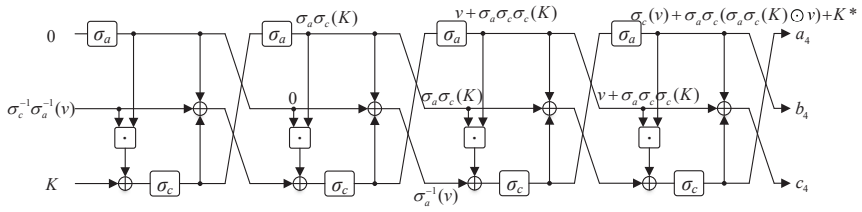


Figure A5 4-round initial structure of Frit$_c^b$-AE, $K^* = \sigma_a\sigma_c(\sigma_a\sigma_c\sigma_c(K) \odot \sigma_a\sigma_c(K))$.

Appendix A.5.1 *Experiments on 11-round Frit$_c^b$-AE.*

Applying the 4-round initial structure in Figure A5 to the 11-round Frit$_c^b$-AE, we can use a 22-dimension cube to get 1 bit condition of K, which is similar to the experiment on 10-round Frit$_a^b$-AE. The three examples of recovering 1 bit condition are listed in Table A8. It is clear that the only differences are the bit conditions, which are recovered by the same cube variables. Recovering 128-bit K needs to solve a set of 128 linear equations, which are also can be calculated by Algorithm A3. Testing about 100 random keys also has a success rate of 100% in about 8 minutes at each.

Table A8 Bit conditions and cube variables of 11-round Frit$_c^b$-AE

Bit conditions	Degree	Cube variables
$K_4 + K_{51} + K_{74} + K_{81} + K_{91}$	22	$v_0, v_1, v_4, v_7, v_8, v_{15}, v_{16}, v_{23}, v_{30}, v_{31}, v_{38}, v_{39}$
$+K_{92} + K_{104} + K_{114} + K_{122}$		$v_{45}, v_{46}, v_{53}, v_{60}, v_{61}, v_{68}, v_{69}, v_{75}, v_{76}, v_{83}$
$K_{10} + K_{33} + K_{40} + K_{50} + K_{51}$	22	$v_0, v_1, v_4, v_7, v_8, v_{15}, v_{16}, v_{23}, v_{30}, v_{31}, v_{38}, v_{39}$
$+K_{63} + K_{73} + K_{81} + K_{91}$		$v_{46}, v_{53}, v_{60}, v_{61}, v_{68}, v_{69}, v_{75}, v_{76}, v_{83}, v_{91}$
$K_{33} + K_{56} + K_{63} + K_{73} + K_{74}$	22	$v_0, v_1, v_4, v_7, v_8, v_{15}, v_{16}, v_{23}, v_{30}, v_{31}, v_{38}, v_{39}$
$+K_{86} + K_{96} + K_{104} + K_{114}$		$v_{46}, v_{53}, v_{60}, v_{61}, v_{68}, v_{69}, v_{75}, v_{76}, v_{83}, v_{114}$

References

1 http://www.gurobi.com/
2 http://www.sagemath.org/