Cryptanalysis of Full Round Fruit

Santanu Sarkar

IIT Madras, Chennai

Outline of the talk

- Fruit Description
- Cryptanalysis of Fruit

Fruit

- Fruit is designed by V. A. Ghafari, H. Hu and Y. Chen in 2017.

Fruit

- Fruit is designed by V. A. Ghafari, H. Hu and Y. Chen in 2017.
- It is a cipher in which the state size is equal to key size.

Fruit

- Fruit is designed by V. A. Ghafari, H. Hu and Y. Chen in 2017.
- It is a cipher in which the state size is equal to key size.
- It uses round key to update the NFSR and also for output keystream.

Fruit

- Fruit is designed by V. A. Ghafari, H. Hu and Y. Chen in 2017.
- It is a cipher in which the state size is equal to key size.
- It uses round key to update the NFSR and also for output keystream.
- Only two attacks has been suggested so far against this cipher.

The birthday paradox

- Consider a random group of 40 people

The birthday paradox

- Consider a random group of 40 people
- What is the probability that someone has the same birthday as you?
- What is the probability that at least two people share the same birthday?

1st Problem

$P(\text{someone has your birthday})$
$= 1 - P(\text{none of the 40 people has your birthday})$
$= 1 - (\frac{364}{365})^{40}$
$= 10.4\%$

2nd Problem

$$
\begin{aligned}
&P(\text{two people have the same birthday}) \\
&= 1 - P(\text{all 40 people have different birthdays}) \\
&= 1 - \tfrac{365}{365} \cdot \tfrac{364}{365} \cdots \tfrac{326}{365} \\
&= 89.1\%
\end{aligned}
$$

Structure

- **Counters:**
  Cr : 7-bit counter $(c_t^0, c_t^1, c_t^2, \cdots, c_t^6)$.
  Cc : 8-bit counter $(c_t^7, c_t^8, c_t^9, \cdots, c_t^{14})$.
  Both these counter starts from 0 and increases by one at each clock. These two counters are independent.

Structure

- **Counters:**
  Cr : 7-bit counter $(c_t^0, c_t^1, c_t^2, \cdots, c_t^6)$.
  Cc : 8-bit counter $(c_t^7, c_t^8, c_t^9, \cdots, c_t^{14})$.
  Both these counter starts from 0 and increases by one at each clock. These two counters are independent.
- K: the secret key $(k_0, k_1, \cdots, k_{79})$.

  Round key function:

$$k_t' = k_s k_{(y+64)} \oplus k_{(u+72)} k_p \oplus k_{(q+32)} \oplus k_{(r+64)}.$$

$s = (c_0^t c_1^t c_2^t c_3^t c_4^t c_5^t), y = (c_3^t c_4^t c_5^t), u = (c_4^t c_5^t c_6^t),$
$p = (c_0^t c_1^t c_2^t c_3^t c_4^t), q = (c_1^t c_2^t c_3^t c_4^t c_5^t), r = (c_3^t c_4^t c_5^t c_6^t).$

- **LFSR:** The LFSR is of 43 bits. $(l_t, l_{t+1}, l_{t+2}, \cdots, l_{t+42})$. Feedback rule:

$$l_{(t+43)} = f(L_t) = l_t \oplus l_{t+8} \oplus l_{(t+18)} \oplus l_{(t+23)} \oplus$$

$$l_{(t+28)} \oplus l_{(t+37)}.$$

- **LFSR:** The LFSR is of 43 bits. $(l_t, l_{t+1}, l_{t+2}, \cdots, l_{t+42})$.
  Feedback rule:

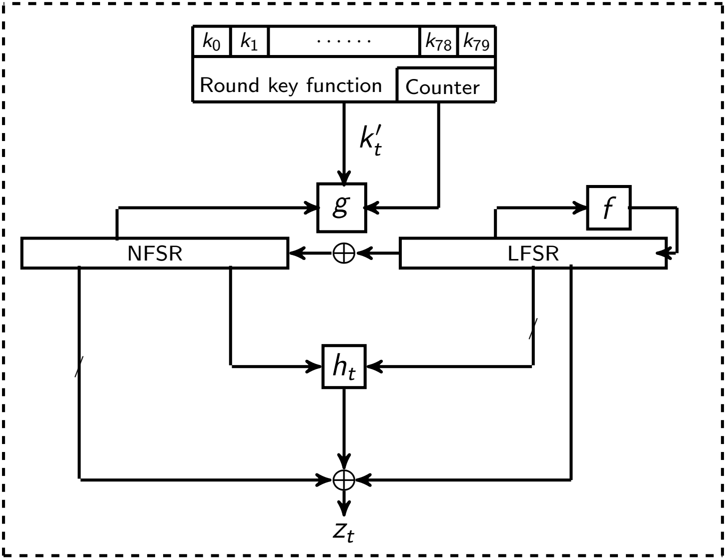
$$l_{(t+43)} = f(L_t) = l_t \oplus l_{t+8} \oplus l_{(t+18)} \oplus l_{(t+23)} \oplus$$

$$l_{(t+28)} \oplus l_{(t+37)}.$$

- **NFSR:** The length of NFSR is 37 bits.$(n_t, n_{t+1}, n_{t+2}, \cdots, n_{t+36})$
  Feedback function:

$$n_{t+37} = {\color{red} g(N_t)} \oplus k'_t \oplus l_t \oplus c_t^{10},$$

  where g is given by
  ${\color{red} g(N_t)} = n_t \oplus n_{t+10} \oplus n_{t+20} \oplus n_{t+12} n_{t+3}$
  $\oplus n_{t+14} n_{t+25} \oplus n_{t+5} n_{t+23} n_{t+31}$
  $\oplus n_{t+8} n_{t+18} \oplus n_{t+28} n_{t+30} n_{t+32} n_{t+34}.$



k_0
k_1
$\cdots\cdots$
k_{78}
k_{79}
Round key function
Counter
k'_t
g
f
NFSR
LFSR
h_t
z_t

Output function:

$$z_t = h_t \oplus n_t \oplus n_{t+7} \oplus n_{t+13} \oplus n_{t+19} \oplus n_{t+24} \oplus n_{t+29} \oplus n_{t+36} \oplus l_{t+38},$$

where

$$h_t = n_{t+1} l_{t+15} \oplus l_{t+1} l_{t+22} \oplus n_{t+35} l_{t+27}$$
$$\oplus n_{t+33} l_{t+11} \oplus l_{t+6} l_{t+33} l_{t+42}.$$

Attack Idea

Attack Idea

Three types of Sieving

- 1-bit Sieving
- Probabilistic Sieving
- Equation satisfaction

1-bit Sieving

$$
\begin{aligned}
z_t = {} & n_{t+1} l_{t+15} \oplus l_{t+1} l_{t+22} \oplus n_{t+35} \\
& l_{t+27} \oplus n_{t+33} l_{t+11} \oplus l_{t+6} l_{t+33} l_{t+42} \\
& \oplus n_t \oplus n_{t+7} \oplus n_{t+13} \oplus n_{t+19} \oplus n_{t+24} \oplus n_{t+29} \oplus n_{t+36} \oplus l_{t+38}.
\end{aligned}
$$

1-bit Sieving

$$z_t = n_{t+1} l_{t+15} \oplus l_{t+1} l_{t+22} \oplus n_{t+35}$$
$$l_{t+27} \oplus n_{t+33} l_{t+11} \oplus l_{t+6} l_{t+33} l_{t+42}$$
$$\oplus n_t \oplus n_{t+7} \oplus n_{t+13} \oplus n_{t+19} \oplus n_{t+24} \oplus n_{t+29} \oplus n_{t+36} \oplus l_{t+38}.$$

At any clock t, if we know the internal state, we can compute the output z_t, without knowing any key bit.

1-bit Sieving

$$z_t = n_{t+1}l_{t+15} \oplus l_{t+1}l_{t+22} \oplus n_{t+35}$$
$$l_{t+27} \oplus n_{t+33}l_{t+11} \oplus l_{t+6}l_{t+33}l_{t+42}$$
$$\oplus n_t \oplus n_{t+7} \oplus n_{t+13} \oplus n_{t+19} \oplus n_{t+24} \oplus n_{t+29} \oplus n_{t+36} \oplus l_{t+38}.$$

At any clock t, if we know the internal state, we can compute the output z_t, without knowing any key bit.

As z_t is already known, this gives us sieving of one bit, i.e, only by knowing 79 bits of the state, we can compute the remaining bit from z_t. So, the number of possible state candidates is reduced by half i.e, 2^{79}.

Probabilistic Sieving

Round key generation:

$$k'_t = k_s k_{y+64} \oplus k_{(u+72)} k_p \oplus k_{(q+32)} \oplus k_{(r+64)}.$$

128 possible counter values for $(c_t^0, c_t^1, c_t^2, \cdots, c_t^6)$.

Probabilistic Sieving

Round key generation:

$$k'_t = k_s k_{y+64} \oplus k_{(u+72)} k_p \oplus k_{(q+32)} \oplus k_{(r+64)}.$$

128 possible counter values for $(c_t^0, c_t^1, c_t^2, \cdots, c_t^6)$.

Bias observed for k'_t:

1. High probability of occurrence of 0.

Probabilistic Sieving

Round key generation:

$$k'_t = k_s k_{y+64} \oplus k_{(u+72)} k_p \oplus k_{(q+32)} \oplus k_{(r+64)}.$$

128 possible counter values for $(c_t^0, c_t^1, c_t^2, \cdots, c_t^6)$.

Bias observed for k'_t:

1. High probability of occurrence of 0.
2. High probability of consecutive bits being equal.

- The following table shows the bias towards 0.

Counter with $\Pr(k'_t = 0) = \frac{3}{8}$	Counter with $\Pr(k'_t = 0) = \frac{5}{8}$
64	72-79
80	88-95
96	104-111
112	120-127

Table: Distribution of k'_t for different counter values.

- there are 32 counter values for which $P(k'_t = k'_{t-1}) = \frac{3}{4}$. Similarly there are 16 counter values for which $P(k'_t = k'_{t-2}) = \frac{9}{16}$.

Attack idea using this bias

- While guessing an r-bit string for $k'_{t-1}, k'_{t-2}, \cdots k'_{t-r}$, we arrange all r-bit string in decreasing order of their probability of occurrence.
- We take our first guess as $\overbrace{00\cdots0}^{r}$, and continue according to the decreasing order of probability.

Reduction factor

Suppose, X_r is a random variable which denotes the number of guesses required to find the correct $k'_{t-1}, \cdots, k'_{t-r}$. Now we denote the expected of value of X_r by $E(X_r)$.

r	6	8	10	12	14
$E(X_r)$	30.777	121.527	484.527	1936.527	7744.526
Reduction factor	2.079	2.107	2.113	2.115	2.116

Table: Reduction factor for different r consecutive guesses.

Final possible states

r	10	12	14	16	18	20
$2^{79-r} \times E(X_r)$	$2^{77.58}$	$2^{77.52}$	$2^{77.46}$	$2^{77.40}$	$2^{77.27}$	$2^{77.08}$

Table: Number of final possible states for different r.

Using our first approach, we have a total of $2^{77.08}$ possible states.

Equation satisfaction

At any time $t_0 > 0$, we guess an 80-bit vector for the internal state (L_{t_0}, N_{t_0}).

Equation satisfaction

At any time $t_0 > 0$, we guess an 80-bit vector for the internal state (L_{t_0}, N_{t_0}).

We have $n_{t+37} = h_{t+1} \oplus n_{t+1} \oplus n_{t+8} \oplus n_{t+14} \oplus n_{t+20} \oplus n_{t+25} \oplus n_{t+30} \oplus z_{t+1} \oplus l_{t+39}$,

Equation satisfaction

At any time $t_0 > 0$, we guess an 80-bit vector for the internal state (L_{t_0}, N_{t_0}).

We have $n_{t+37} = h_{t+1} \oplus n_{t+1} \oplus n_{t+8} \oplus n_{t+14} \oplus n_{t+20} \oplus n_{t+25} \oplus n_{t+30} \oplus z_{t+1} \oplus l_{t+39}$,

Again,

$$n_{t+37} = g(N_t) \oplus k'_t \oplus l_t \oplus c_t^{10}$$

For any $t \geq t_0$,

Equation satisfaction

At any time $t_0 > 0$, we guess an 80-bit vector for the internal state (L_{t_0}, N_{t_0}).

We have $n_{t+37} = h_{t+1} \oplus n_{t+1} \oplus n_{t+8} \oplus n_{t+14} \oplus n_{t+20} \oplus n_{t+25} \oplus n_{t+30} \oplus z_{t+1} \oplus l_{t+39}$,

Again,

$$n_{t+37} = g(N_t) \oplus k'_t \oplus l_t \oplus c_t^{10}$$

For any $t \geq t_0$, $k_s k_{(y+64)} \oplus k_{(u+72)} k_p \oplus k_{(q+32)} \oplus k_{(r+64)} \oplus \alpha_t = 0$. On average, 24 equations sieves 50% of the wrong states.

Preprocessing

- Construction of table $T_1, T_2, \cdots T_l$.
- Each table contains 2^{24} possible output keystream for $\alpha_t, \alpha_{t+1}, \cdots \alpha_{t+23}$.
- For each possible string, possible counter values are stored.
- So, total size of each table is 2^{31}.

Processing

- From the output keystream bit, calculate α_i's.
- Match α_0 to α_{23} with the corresponding string in table 1.
- If there is any counter value available, go to table 2, otherwise discard the state.
- In table 2, again do same for α_{24} to α_{47} and take the intersection of the possible counters for table 1 and 2.
- If intersection is ϕ, discard the state. Otherwise go to next table and repeat.

Final complexity

Final complexity

- Compared to atleast 210 iterations for exhaustive search, using our method, we can eliminate a wrong state from average 48 output bits.

Final complexity

- Compared to atleast 210 iterations for exhaustive search, using our method, we can eliminate a wrong state from average 48 output bits.
- This is equivalent to exhaustive search over $2^{74.95}$ state.

Final complexity

- Compared to atleast 210 iterations for exhaustive search, using our method, we can eliminate a wrong state from average 48 output bits.
- This is equivalent to exhaustive search over $2^{74.95}$ state.

Version 2 of Fruit Using same idea, we can attack the improved second version of Fruit with complexity around 2^{77} This is the only attack proposed so far against the second version.

Plantlet

- 61 bit LFSR
- 40 bit NFSR

Thank You