Cryptanalysis of the Full Spritz Stream Cipher

Subhadeep Banik[1,2] and Takanori Isobe[3]

[1]School of Physical and Mathematical Sciences, NTU
[2]DTU Compute, Technical University of Denmark, Lungby
[3] Sony Corporation, Tokyo, Japan

March 20, 2016

Fast Software Encryption, 2016.

NANYANG TECHNOLOGICAL UNIVERSITY

RC4 Stream Cipher

- Designed by Ron Rivest in 1987.
- Trade Secret till 1994, till source code was leaked on `cypherpunks.com`.
- Used in SSL/TLS, WEP, Microsoft Lotus and Oracle secure SQL.
- Simple and Elegant structure, high speed in software.
- Intense Scrutiny over the last 20 years.

RC4 Cryptanalysis

- Mantin and Shamir: Second Byte Bias [FSE 2001]

$$Pr[Z_2 = 0] = \frac{2}{N}$$

- Improved by Maitra et al (Bias for Z_3 to Z_{256}) [FSE 2011]

$$Pr[Z_r = 0] = \frac{1}{N} + \frac{c_r}{N^2}, \quad \forall\ 3 \leq r \leq 256$$

- State Recovery by Maximov/Khovratovich [CRYPTO 2008].
- Practical plaintext recovery attacks by Alfardan et al. [USENIX 2013]/ Isobe et al [FSE 2013].
- RC4 slowly being phased out of practical use.

Spritz

- Rivest and Schulz in the CRYPTO rump session last year proposed Spritz.
- The designer searched over a million candidate ciphers.
- Experiments took 5 "core-months" of CPU time.
- Basic Stream Cipher mode can be modified to provide functionalities for MAC, Hash Function.
- Slower than RC4 but a good drop in replacement for RC4.

Modules in SPRITZ

INITIALIZESTATE(N)

1. $i = j = k = a = z = 0$, $w = 1$.
2. **for** $v \rightarrow 0$ to $N - 1$
   $S[v] = v$

ABSORB(I)

1. **for** $v \rightarrow 0$ to $I.length - 1$
   ABSORBBYTE($I[v]$)

ABSORBBYTE(b)

1. ABSORBNIBBLE($low(b)$)
2. ABSORBNIBBLE($high(b)$)

ABSORBNIBBLE(x)

1. **if** $a = \lfloor \frac{N}{2} \rfloor$
   SHUFFLE()
2. SWAP($S[a]$, $S[\lfloor N/2 \rfloor + x]$)
3. $a = a + 1$

ABSORBSTOP()

1. **if** $a = \lfloor \frac{N}{2} \rfloor$
   SHUFFLE()
2. $a = a + 1$

SHUFFLE()

1. WHIP($2N$)
2. CRUSH()
3. WHIP($2N$)
4. CRUSH()
5. WHIP($2N$)
6. $a = 0$

WHIP(r)

1. **for** $v \rightarrow 0$ to $r - 1$
   UPDATE()
2. **do** $w = w + 1$
   until $gcd(w, N) = 1$

CRUSH()

1. **for** $v \rightarrow 0$ to $\lfloor N/2 \rfloor - 1$
   if $S[v] > S[N - 1 - v]$
   SWAP($S[v]$, $S[N - 1 - v]$)

SQUEEZE(r)

1. **if** $a > 0$
   SHUFFLE()
2. $P = Array.New(r)$
3. **for** $v \rightarrow 0$ to $r - 1$
   $P[v] =$ DRIP()
4. **return** P

DRIP()

1. **if** $a > 0$
   SHUFFLE()
2. UPDATE()
3. **return** OUTPUT()

UPDATE()

1. $i = i + w$
2. $j = k + S[j + S[i]]$
3. $k = i + k + S[j]$
4. SWAP($S[i]$, $S[j]$)

OUTPUT()

1. $z = S[j + S[i + S[z + k]]]$
2. **return** z

Figure: Modules for Spritz. When N is a power of 2, the last two lines of WHIP are equivalent to $w = w + 2$.

Some Observations

- In stream cipher mode: the sequence of execution is

  A. ABSORB(K)
  B. ABSORBSTOP(), ABSORB(IV) (**optional, if IV is used**)
  C. SQUEEZE().
- "a" plays NO role in the SQUEEZE phase when the cipher produces keystream bytes.
- "w" is constant during the SQUEEZE phase: depends on the length of Key/IV.
- If length of Key is less than $\lfloor N/4 \rfloor$ bytes then $w = 7$.
- Easy to see that at the beginning of SQUEEZE phase: $i = 0$ always.

NANYANG TECHNOLOGICAL UNIVERSITY

Some Observations

- The sequence of updates during the SQUEEZE phase is therefore given as:

SPRITZ

1. $i = i + w$
2. $j = k + S[j + S[i]]$
3. $k = k + i + S[j]$
4. SWAP $(S[i], S[j])$
5. $z = S[j + S[i + S[z + k]]]$

RC4

1. $i = i + 1$
2. $j = j + S[i]$
3. SWAP $(S[i], S[j])$
4. **return** $z = S[S[i] + S[j]]$

- In SPRITZ, next keystream byte depends on current keystream byte.

Main Theorem

Theorem

$\Pr[Z_1 = Z_2 = -w] = \frac{1}{N^2} + \frac{3}{N^4}$.

Proof.

We outline three mutually exclusive events **I, II** and **III**, each of which occurs with probability $\frac{1}{N^4}$, that guarantees that the first two output bytes produced by the cipher are both equal to $-w$.

I. $S[w] = -w,\ S[2w] = 0, k = 0, S[j - w] = 2w$

II. $k = 2w,\ S[j + S[w]] = -2w,\ S[2w] = w, S[0] = -w$

III. $k + S[j - w] = 2w,\ k + S[2w] = 0,\ S[w - k] = 0,\ S[w] = -w$

Main Theorem

Theorem

$\Pr[Z_1 = Z_2 = -w] = \frac{1}{N^2} + \frac{3}{N^4}$.

Proof.

For example, when **I** occurs in the first round we have the following changes :

1. $i \leftarrow i + w = w$
2. $j \leftarrow 0 + S[j + S[w]] = S[j - w] = 2w$
3. $k \leftarrow k + i + S[j] = 0 + w + S[2w] = 0 + w + 0 = w$
4. $S[w] \leftarrow 0,\ S[2w] \leftarrow -w$ after SWAP
5. $z \leftarrow S[2w + S[w + S[w]]] = S[2w + S[w]] = S[2w] = -w$

Main Theorem

Theorem

$\Pr[Z_1 = Z_2 = -w] = \frac{1}{N^2} + \frac{3}{N^4}$.

Proof.

Similarly in the second round we have the following changes:

1. $i \leftarrow i + w = 2w$,
2. $j \leftarrow w + S[2w + S[2w]] = w + S[w] = w$
3. $k \leftarrow k + i + S[j] = w + 2w + S[w] = 3w + 0 = 3w$
4. $S[w] \leftarrow -w$, $S[2w] \leftarrow 0$ after SWAP
5. $z \leftarrow S[w + S[2w + S[3w - w]]] = S[w + S[2w + S[2w]]] = S[w] = -w$

We get similar results when we analyze **II** and **III**. □

□

NANYANG TECHNOLOGICAL UNIVERSITY

Main Theorem

Theorem

$\Pr[Z_1 = Z_2 = -w] = \frac{1}{N^2} + \frac{3}{N^4}$.

Proof.

Let us now denote by E the union of the events **I, II** and **III**. We have $\Pr[\mathrm{E}] = \frac{3}{N^4}$, and $\Pr[Z_1 = Z_2 = -w|\mathrm{E}] = 1$. We assume that when E does not occur $\Pr[Z_1 = Z_2 = -w|\mathrm{E}^c] = \frac{1}{N^2}$, and is more or less uniformly random.

$$\Pr[Z_1 = Z_2 = -w] = \Pr[Z_1 = Z_2 = -w|\mathrm{E}] \cdot \Pr[\mathrm{E}] + \Pr[Z_1 = Z_2 = -w|\mathrm{E}^c] \cdot \Pr[\mathrm{E}^c]$$
$$= 1 \cdot \frac{3}{N^4} + \frac{1}{N^2} \cdot \left[1 - \frac{3}{N^4}\right] \approx \frac{1}{N^2} + \frac{3}{N^4}$$

□

By $\mathcal{O}(\frac{1}{pq^2})$ rule around $\frac{N^6}{9} = 2^{44.8}$ samples required for broadcast attack.

Experimental Verification for small N

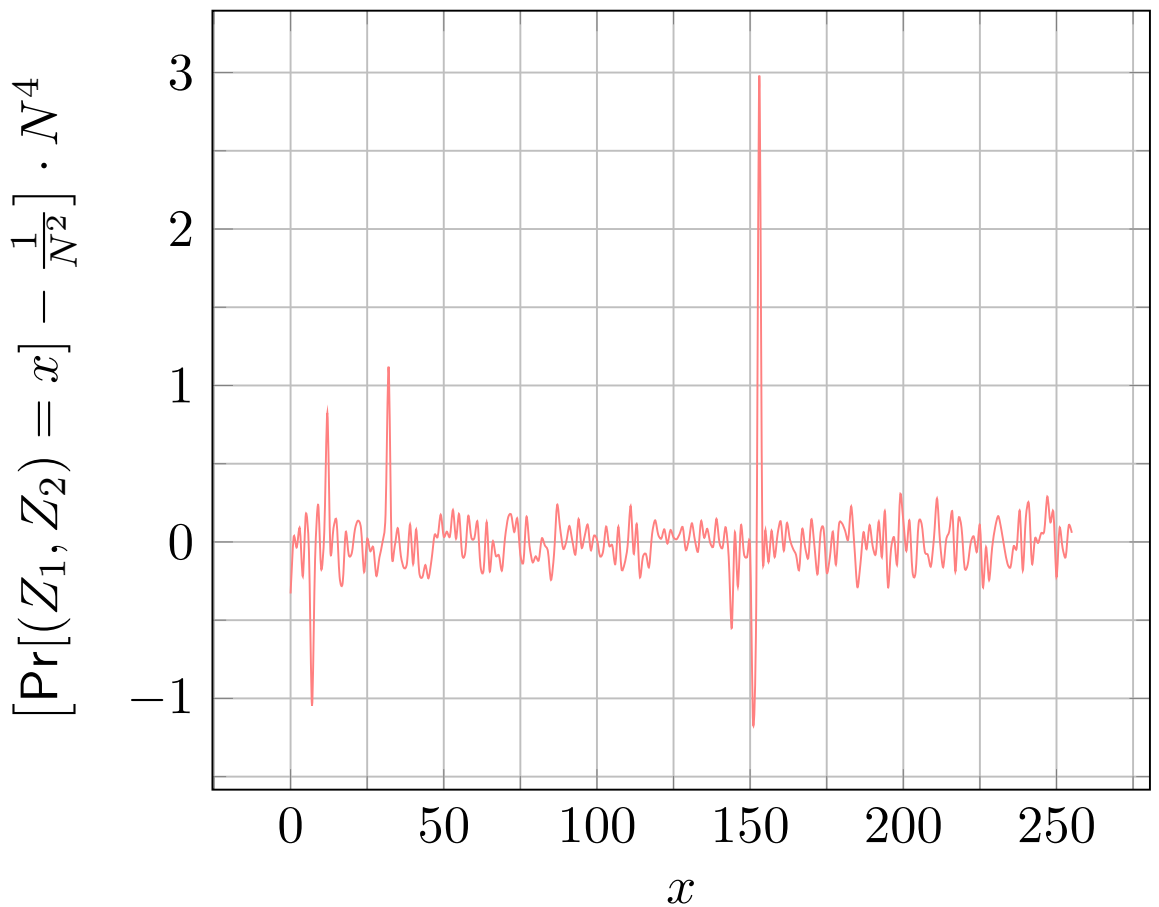


Figure: $\left[\Pr[(Z_1, Z_2) = x] - \frac{1}{N^2}\right] \cdot N^4$ (for $N = 16$)

Experimental Verification for small N

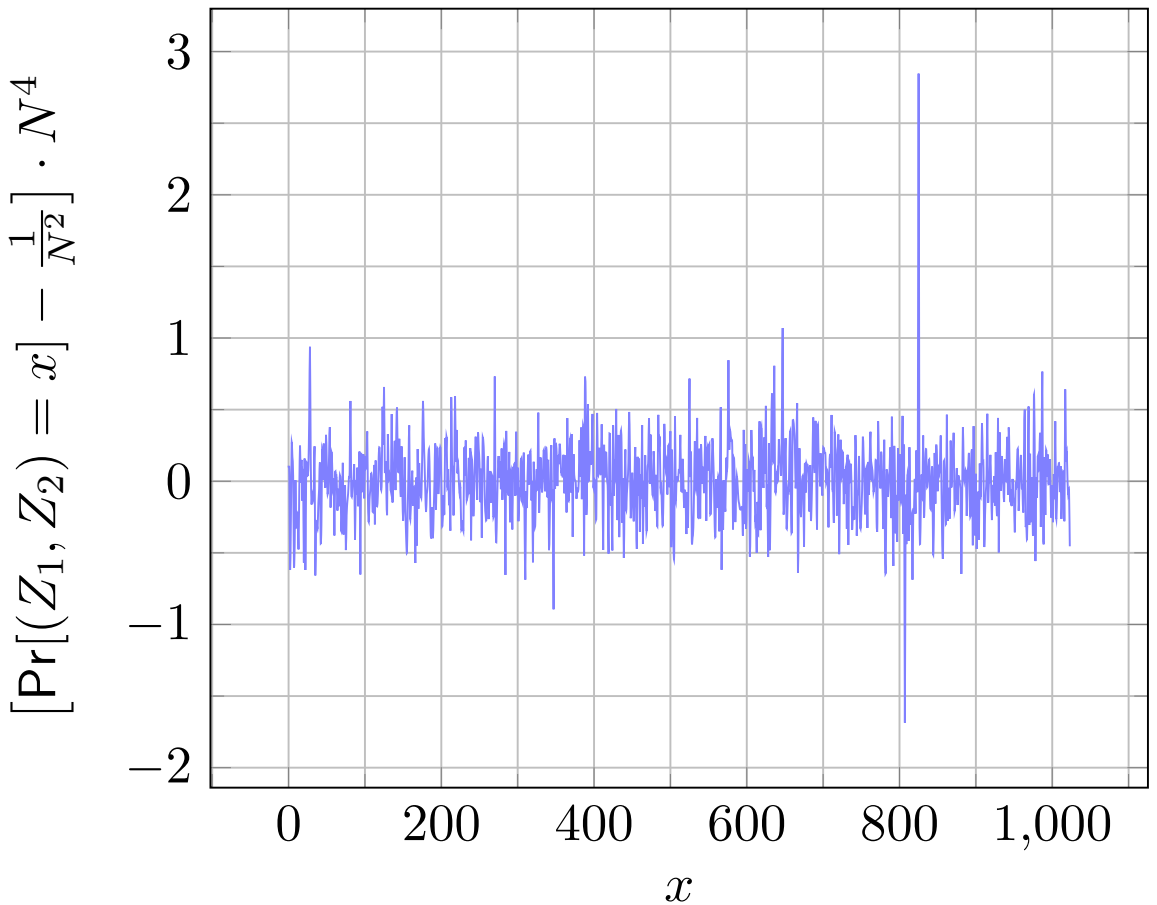


Figure: $\left[\Pr[(Z_1, Z_2) = x] - \frac{1}{N^2}\right] \cdot N^4$ (for $N = 32$)

Keystream Distinguisher for Single Key/IV

- The event $Z_t = Z_{t+1} = -w$ is not biased for arbitrary values of t.
- Previous proof requires $i = z = 0$ at the beginning of the SQUEEZE phase.
- i becomes 0 after N rounds: so we consider the event

$$\Pr[Z_{kN+1} = Z_{kN+2} = -w | Z_{kN} = 0]$$

- This probability is also $\frac{1}{N^2} + \frac{3}{N^4}$.
- Note $Z_{KN} = 0$, once every N cycles
- Total keystream required is $N \cdot N \cdot \frac{N^6}{9} \approx 2^{60.8}$

State Recovery Attack

- State recovery of RC4 : Maximov/Khovratovich [CRYPTO 2008] takes 2^{241} steps.
- Spritz state : $N! \cdot N^6 \approx 2^{1732}$
- At Latincrypt 2015: Ankele/Kölbl/Rechberger did a similar analysis on SPRITZ
  - State recovery of SPRITZ using 3 approaches.
  - Simple Guess and Determine was found to be the best.
  - Requires 2^{1400} steps.
  - Much more secure than simple RC4.
- We do a state recovery attack in the Single Key/Multiple IV mode.

SPECIAL State

Definition

Define a Spritz state as the 3-tuple (S, j, k) just at the beginning of the SQUEEZE phase. A Spritz state is called a SPECIAL state if all the following conditions hold simultaneously.

1. $S[t] \equiv 0 \bmod 2$, if $t \equiv 1 \bmod 2$,
2. $S[t] \equiv 1 \bmod 2$, if $t \equiv 0 \bmod 2$,
3. $j \equiv 0 \bmod 2$ and $k \equiv 0 \bmod 2$

Keystream with Periodic LSB

Lemma

If the state at the beginning of the SQUEEZE *phase is a* SPECIAL *state then the following hold (assuming N is even):*

a) *The state after every four iterations is a* SPECIAL *state.*

b) *In every iteration, the updated values of i and j are equal modulo 2. Hence no* SWAP *between odd and even values occur. And so, even and odd indexed positions of the S array will continue to hold odd and even values respectively.*

c) $Z_t \equiv Z_{t+4} \bmod 2$, *for all values of t.*

Keystream with Periodic LSB

Proof.

- i and z are 0 and so even at the beginning.
- If N is even, the design of the WHIP module ensures that the value of w is odd.

#	Index	$t = 1$	$t = 2$	$t = 3$	$t = 4$
1	$i = i + w$*	1	0	1	0
2	$j + S[i]$*	0	0	0	0
3	$j = k + S[j + S[i]]$*	1	0	1	0
4	$k = k + i + S[j]$	1	0	1	0
5	$z + k$*	1	0	0	1
6	$i + S[z + k]$*	1	1	0	0
7	$j + S[i + S[z + k]]$*	1	0	0	1
8	$z = S[j + S[i + S[z + k]]]$	0	1	1	0

Table: The modulo 2 values of the various indices through 4 iterations.

Keystream with Periodic LSB

Proof.

- After 4 rounds i, j, k, z become even again and cycle repeats.
- The modulo 2 values of the keystream byte z becomes periodic with period 4: $0, 1, 1, 0, \quad 0, 1, 1, 0, \quad 0, 1, 1, 0 \ldots$

#	Index	$t = 1$	$t = 2$	$t = 3$	$t = 4$
1	$i = i + w$*	1	0	1	0
2	$j + S[i]$*	0	0	0	0
3	$j = k + S[j + S[i]]$*	1	0	1	0
4	$k = k + i + S[j]$	1	0	1	0
5	$z + k$*	1	0	0	1
6	$i + S[z + k]$*	1	1	0	0
7	$j + S[i + S[z + k]]$*	1	0	0	1
8	$z = S[j + S[i + S[z + k]]]$	0	1	1	0

Table: The modulo 2 values of the various indices through 4 iterations.

Probability of SPECIAL State

Combinatorially, it is easy to see that the probability of SPECIAL states is:

$$\rho = \frac{\left(\frac{N}{2}\right)^2 \cdot \left[\left(\frac{N}{2}\right)!\right]^2}{N^2 \cdot (N!)}$$

For $N = 256$, $\rho \approx 2^{-253.7}$.

1. For a fixed key, and Multiple IVs collect keystream of around $10 * N$ bytes and inspect the sequence Z_t mod 2.
2. If the sequence is $0, 1, 1, 0, \quad 0, 1, 1, 0, \quad 0, 1, 1, 0 \ldots$ i.e. periodic with period 4, then the attacker can conclude with high probability that he has encountered a SPECIAL state and he proceeds according to Guess and Determine Algorithm.
3. The above technique is likely to succeed once in ρ^{-1} attempts.

Guess and Determine: Highlights

- Assign odd/even values to even/odd indices and evaluate each guess.
- Compute the index $d = j + S[i + S[z + k]]$ with the guessed values.
- Performs the Verification step: Depending on the comparison between $S[d]$ and the current keystream byte Z_r:

If $S[d] = \text{NULL}$ and $Z_r \notin S \rightarrow$ Assign $S[d] = Z_r$, Go to next round $r + 1$

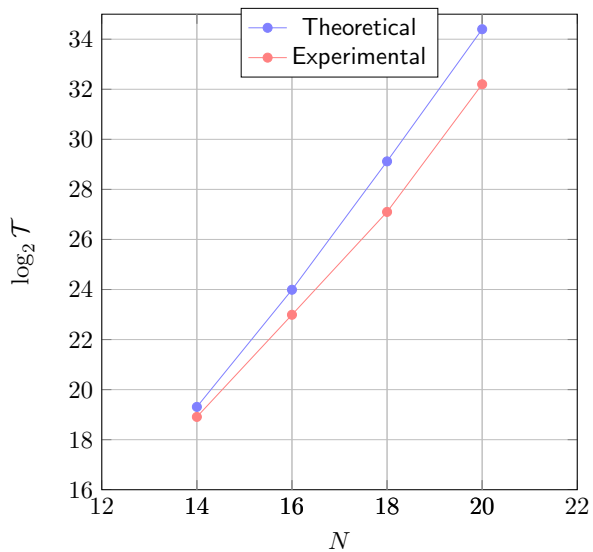
If $S[d] = \text{NULL}$ and $Z_r \in S \rightarrow$ Contradiction!! Try another assignment

If $S[d] \neq \text{NULL}$ and $Z_r \neq S[d] \rightarrow$ Contradiction!! Try another assignment

If $S[d] \neq \text{NULL}$ and $Z_r = S[d] \rightarrow$ Go to next round $r + 1$

Theoretical vs Experimental

- Experiments done for $N = 14, 16, 18, 20$.
- Theoretical complexity overestimates the experimentally obtained values.



- By theoretical estimates, for $N = 256$, one should require approximately 2^{1247} steps

NANYANG TECHNOLOGICAL UNIVERSITY

Summary of Results

	Type of Attack	Complexity	Reference
1	Distinguishing attack on scaled down version ($N = 8$)	$2^{21.9}$ outputs	Zoltak [2014/985]
2	Distinguishing attack on full Spritz in multiple key-IV setting	$2^{44.8}$ outputs	This Work
3	Distinguishing attack on full Spritz in single key-IV setting	$2^{60.8}$ outputs	This Work
4	State recovery attack	2^{1400} steps	Ankele et al.[Latincrypt 15]
		2^{1247} steps	This Work

Table: Summary of Results on Spritz

Conclusion and Possible Research Directions

- New word based cipher for RC4 replacement: SPEED should be faster/comparable to RC4.
- Long term BIAS for SPRITZ.
- Finding collisions in the HASH function mode of SPRITZ.
- More efficient state recovery of SPRITZ.

THANK YOU

Questions??