Some observations on ACORN v1 and Trivia-SC

Rebhu Johymalyo Josh
Chennai Mathematical Institute,
SIPCOT IT Park, Siruseri,
Chennai- 603103, India
rebhu.webs@gmail.com

Santanu Sarkar
Department of Mathematics,
Indian Institute of Technology Madras,
Chennai - 600036, India.
sarkar.santanu.bir@gmail.com

Lightweight Cryptography Workshop 2015,
National Institute of Standards and Technology,
Gaithersburg, USA

21st day of July, 2015

Acorn v1

Initialization

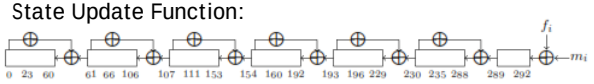
Index of message bit (m)	-1536 to -1536 + 127	-1408 to -1408 + 127	-1280	-1279 to 0
Value	KEY (128-bit)	IV (128-bit)	1	0

Index of control bits (ca & cb)	-1536 to -1
Value	1

Initial State (S_{-1536}) = 0, where S is a 292-bit variable

Update state for 1535 times by running StateUpdateFunction(m,ca,cb) which alters a few bits of the previous state.

State Update Function:



Acorn v1

Processing Associated Data

Index of message bit (m)	0 to length of associated data(adlen) -1	Adlen	Adlen+1 to adlen+ 511
Value	Value of associated data at that index	1	0

Index of control bits (ca & cb)	0 to adlen + 255	adlen + 256 to adlen + 511
Value (ca)	1	0
Value (cb)	1	1

Update state for 511 + adlen times by running StateUpdateFunction(m,ca,cb) which alters a few bits of the previous state.

Acorn v1

Encryption of Data

Index of message bit (m)	adlen + 512 to adlen + 512 + datalen 1	adlen + 512 + datalen	adlen + 512 + datalen +1 to adlen + 1023 + datalen
Value	Value of message bit at that index	1	0

Index of control bits (ca & cb)	adlen + 512 to adlen + 767 + datalen	adlen + 768 + datalen to adlen + 1023 + datalen
Value (ca)	1	0
Value (cb)	1	1

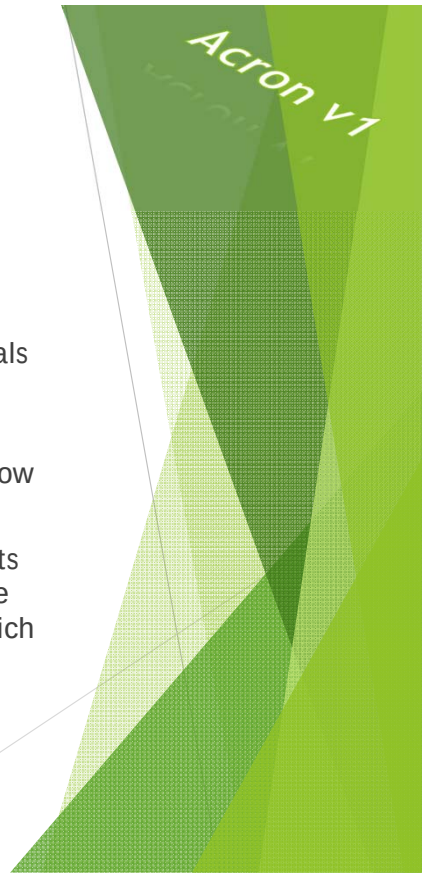
Update state for 512 + datalen times by running StateUpdateFunction(m,ca,cb) which alters a few bits of the previous state.
Then plaintext bit is XOR with KSG bit to return cipher text

$$KSG_i = S_{i,12} \oplus S_{i,54} \oplus (S_{i,235} \oplus S_{i,61}) \oplus (S_{i,235} \oplus S_{i,193}) \oplus (S_{i,61} \oplus S_{i,193})$$

Cube Attack

A brief overview

- Published by Itai Dinur and Adi Shamir in September 2008
- Almost any cryptographic scheme can be described by tweakable polynomials over GF(2), which contain both secret variables (e.g., key bits) and public variables (e.g., plaintext bits or IV bits).
- A cipher is vulnerable if an output bit can be represented as a sufficiently low degree polynomial over GF(2) of key and input bits.
- In stream ciphers the output depends on secret key bits and on public IV bits which can be chosen arbitrarily. By modifying the values of these tweakable public bits, the attacker can obtain many derived polynomial equations which are closely related.



Our Attempt of Attack

Acron v1

- A fixed message
- Regenerate cipher text for 64000 random keys and for each key, every associated data bits possible.
- At specific index, all the output bits obtained at that index is XORed.

Result of the attack

Acron v1

- XORed value obtained at each index is 0 till some upper bound dependent on the length of associated data.
- By different extrapolation, we inferred that 0 to 512 bits' XOR value will be 0 if the length of associated data varies from 37 to 45 approximately.
- So to prevent as predictable XOR value, StateUpdate round of 1024 will be safer (by experiment and also by extrapolation)

Length of associated data	Upper bound till which value obtained is 0
10	341
12	349
14	358
16	371
18	379

TriviA-ck-v1

Keylogging Algorithm

- $(A_1, \dots, A_{132}) = (k_1, \dots, k_{128}, 1, 1, 1, 1)$
- $(B_1, \dots, B_{105}) = (1, \dots, 1)$
- $(C_1, \dots, C_{147}) = (v_1, \dots, v_{128}, 1, \dots, 1)$

TriviA-ck-v1

Keystream Algorithm and Pseudo-Random Generation Algorithm

- $t_1 \leftarrow A_{66} \oplus A_{132} \oplus (A_{130} \wedge A_{131}) \oplus B_{96}$
- $t_2 \leftarrow B_{69} \oplus B_{105} \oplus (B_{103} \wedge B_{104}) \oplus C_{120}$
- $t_1 \leftarrow C_{66} \oplus C_{147} \oplus (C_{145} \wedge C_{146}) \oplus A_{75}$
- $(A_1, \ldots, A_{132}) \leftarrow (t_3, A_1, \ldots, A_{132})$
- $(B_1, \ldots, B_{105}) \leftarrow (t_1, B_1, \ldots, B_{104})$
- $(C_1, \ldots, C_{147}) \leftarrow (t_2, C_1, \ldots, C_{146})$
- $z_i \leftarrow A_{66} \oplus A_{132} \oplus B_{69} \oplus B_{105} \oplus C_{66} \oplus C_{147} \oplus (A_{102} \wedge B_{66})$

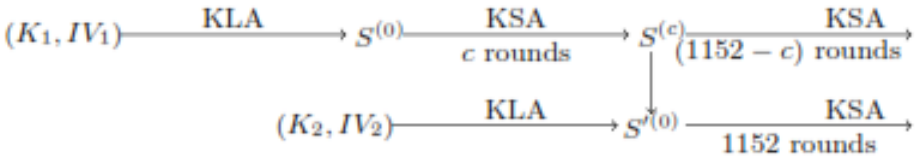
TriviA-ck-v1

Working

- There are 3 Non-Linear Feedback Shift Register namely A, B and C with 132, 105, 147 bits respectively.
- There is a internal state register of 384 bits (denoted by S)
- The assignment of NFSRs are as follows:-
  - $A \equiv (S_1, \dots, S_{132})$
  - $B \equiv (S_{133}, \dots, S_{237})$
  - $C \equiv (S_{238}, \dots, S_{384})$
- The evolution is done for 1152 rounds before generation of key stream bit

The Attack

TriviA-ck-v1



- Total 256 variables of 128 key and 128 IV.
- Now after c rounds of KSA, if the state S^c satisfies 384-256 = 128 fixed locations of S^0, then S^c will be a starting state of a different pair of key and IV.

TriviA-ck v1

Previous Attack

TriviA-ck-v1

- Initial state $S \equiv (A, B, C)$
- $A = (k_1, \dots, k_{128}, 1, 1, 1, 1)$
- $B = (1, 1, \dots, 1)$
- $C = (v_1, \dots, v_{128}, 1, \dots, 1)$
- After shifting the clock once, the states are:
  - $A^1 = (t_3, k_1, \dots, k_{128}, 1, 1, 1)$
  - $B^1 = (t_1, 1, \dots, 1)$
  - $C^1 = (t_2, v_1, \dots, v_{128}, 1, \dots, 1)$
- If $k_{128} = v_{128} = t_1 = 1$, then $S^1 \equiv (A^1, B^1, C^1)$ is a valid state generated by the key $(t_3, k_1, \dots, k_{127})$ and IV $(t_2, v_1, \dots, v_{127})$

The Attack ... continued

TriviA-ck-v1

- We have 128 equations over 256 variables.
- Used SAT Solver
- In some cases, S^c can be really complicated. So we have introduced 3 new variables at each stage of KSA namely x_i, y_i, z_i to replace t_1, t_2, t_3.Thus 3 new equations are introduced at each stage.
- Total equations are 128+3c over 256+3c variables.
- We can expected to find a solution as there are more variables than equations.

Result of the Attack

TriviA-ck-v1

- K_1: 09c2e824283614f6034c2fee86e2e9b6
- IV_1 : e6cf0aac80f27ea07436c1c05137582b
- Key-streams:
  $b3425795e81caa3a6d5e934a464427c7251748080a7e50$
  $bdb3a0de00196662eff03370\ \overbrace{484d089cebca7e28e90cfe0}$
- K_2 : 2c41a48c695055f80c6b23d4c5c9db96
- IV_2 : cf7040af7c63795f0d254746d25c778b
- Keysteams:
  $\overbrace{484d089cebca7e28e90cfe0}\ 85926d614a476dca7c1424$

Thank you!