Finding Slid Pairs for the Plantlet Stream Cipher

Joshua Copeland and **Leonie Simpson**

Queensland University of Technology

lr.simpson@qut.edu.au

February 5, 2020

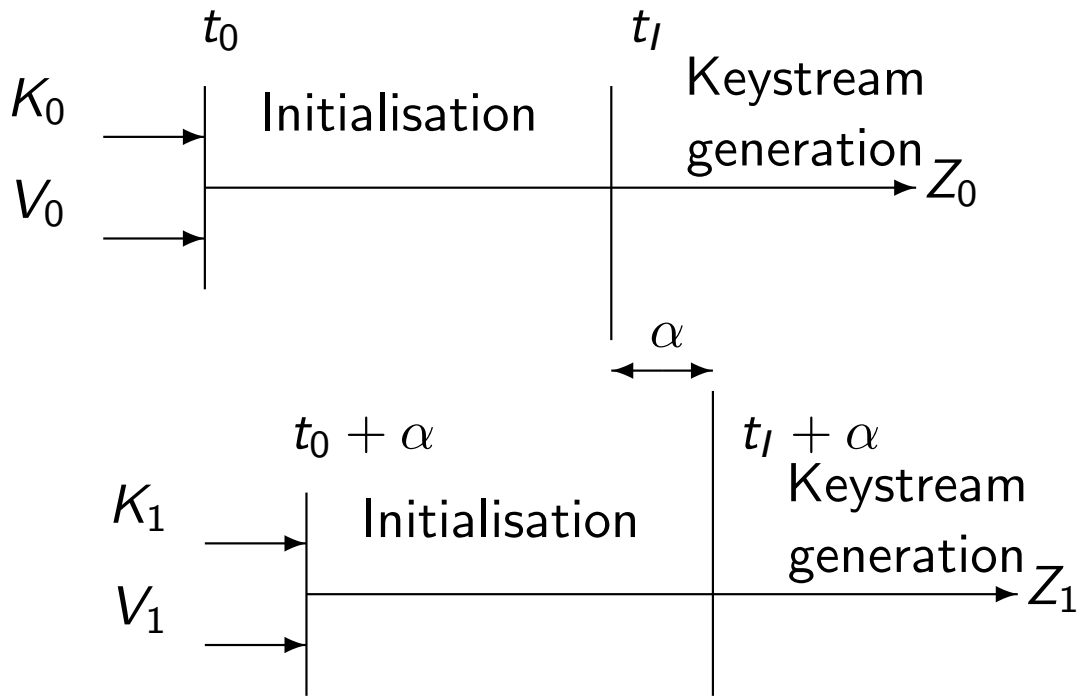
Overview

Plantlet Stream Cipher

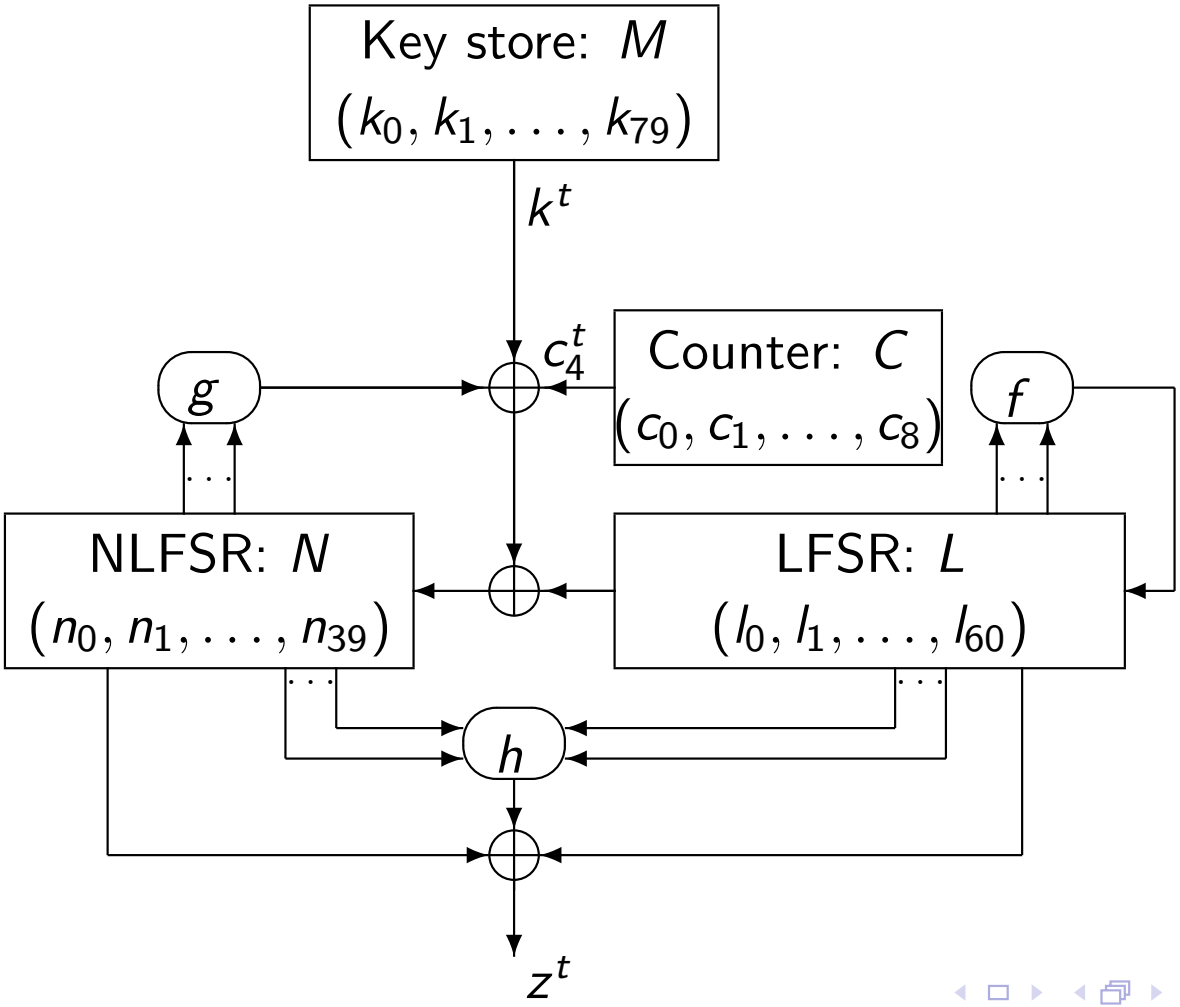
- Lightweight binary additive stream cipher proposed in 2016
  - Mikhalev, Armnecht and Muller
- Keystream generator design based on bit-wise shift registers
  - Similar to Grain, Sprout
- Two inputs:
  - 80-bit *secret* key
  - 90-bit *known* initialisation vector (IV)
- Output: a binary sequence of length $\leq 2^{30}$ bits
  - Encryption: XOR this sequence with plaintext
  - Decryption: XOR this sequence with ciphertext

What are Slid Pairs?

- Different (key, IV) pairs that produce phase-shifted versions of the same binary sequence
- Should be hard to find slid pairs or determine relationships



Plantlet Structure



Plantlet Operation - Modes

Initialisation

- **Loading phase**:
  - First 40 bits of IV loaded into (40-bit) NLFSR
  - Remaining 50 bits of IV loaded into (61-bit) LFSR
  - Remaining LFSR stages loaded with 1's - except for one 0
- **Diffusion phase**:
  - State update function applied to internal state 320 times
  - Register feedback, key bit k^t and counter bit c^t used in update
  - No keystream output BUT z^t used in both LFSR and NLFSR updates

Keystream generation

- z^t used as keystream only - not in LFSR or NLFSR feedback

Observations: Periodic Subsequences

Key component

- Plantlet makes continuous use of the key - throughout initialisation and keystream generation
- One key bit used as input to NLFSR update at each time step
- $k^t = k_{t \bmod 80}$; periodic with period = 80 or a divisor of 80

Counter component

- 7 bits of C used for simple counter: from 0 to 79 then reset
- Stage 4 content forms input to NLFSR update each time step
- Counter component produces a fixed binary sequence
- $C_4 = (0^{16}1^{16}0^{16}1^{16}0^{16})$; periodic with period = 80

Observations: Register Autonomy?

LFSR component

- During **initialisation**, z^t used in LFSR update
- During **keystream generation**, LFSR is autonomous
  - Primitive feedback function and register contents not all-zero at end of initialisation, so LFSR output is binary sequence with period $2^{61} - 1$

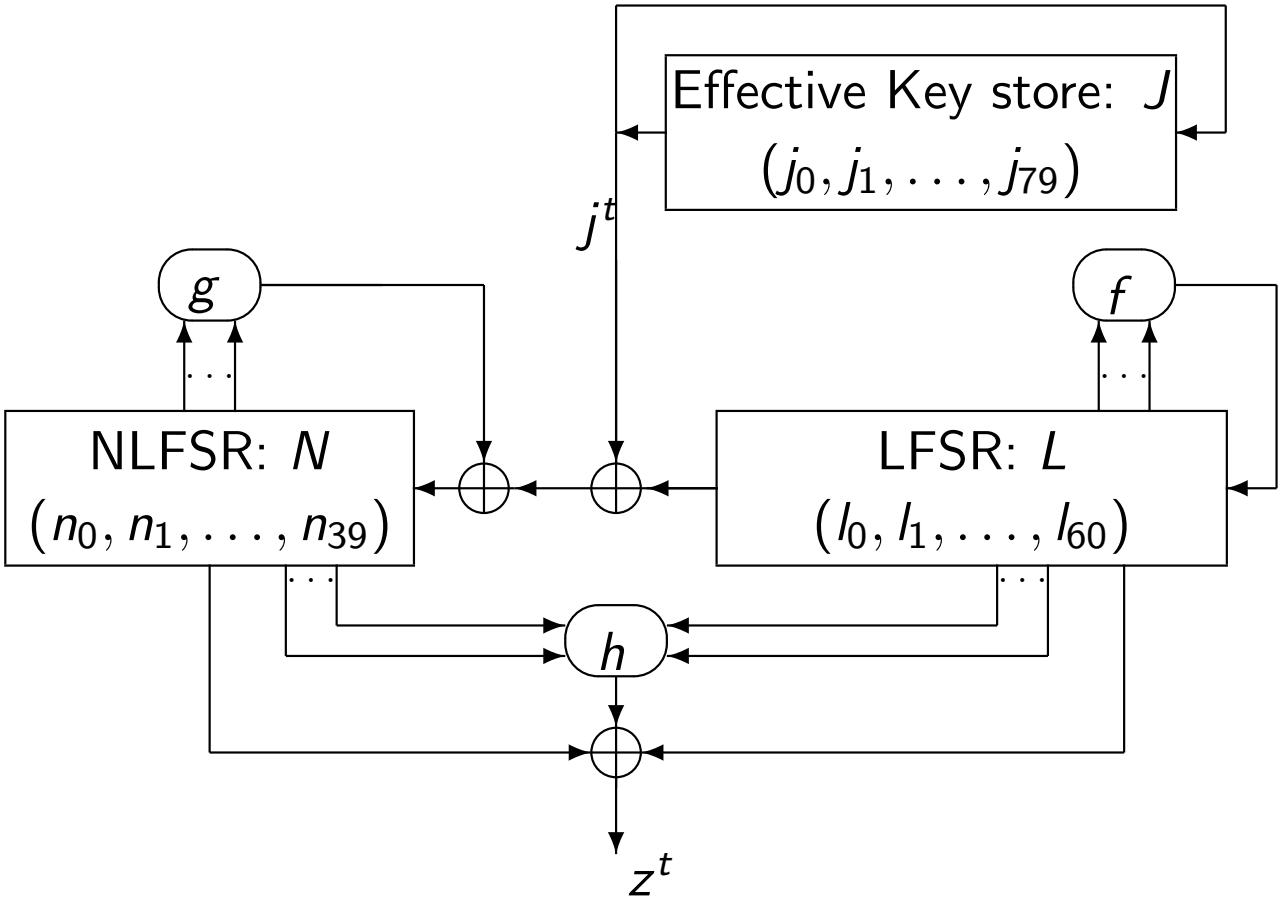
NFSR component

- NFSR not autonomous in either initialisat'n or keystream gen
- During **initialisation**, $n_{39}^{t+1} = g(N^t) \oplus z^t \oplus l_0^t \oplus c_4^t \oplus k^t$
- During **keystream generation**, $n_{39}^{t+1} = g(N^t) \oplus l_0^t \oplus c_4^t \oplus k^t$

Can we simplify Plantlet?

- During initialisation and keystream generation, both M and C
  - Are autonomous components
  - Produce sequences with period 80, or a divsor of 80
- Outputs of M & C (k^t & c_4^t) XORed in NLFSR state update
- Combine both M and C into single component J
  - J produces sequence with period 80 (or a divisor of 80)
  - Adjust value of Plantlet key K by combining with 80-bit counter sequence C_4
  - Effective key: $K \oplus C_4$
- Simplified Plantlet using $K \oplus C_4$ produces same keystream as Plantlet with K
- Simpler design is easier to analyze
  - Acknowledgement of Micah Brown investigation, 2018

Simplified Plantlet Structure



Initialisation

- **Loading phase - same as Original Plantlet**:
  - First 40 bits of IV loaded into (40-bit) NLFSR
  - Remaining 50 bits of IV loaded into (61-bit) LFSR
  - Remaining LFSR stages loaded with 1's - except for one 0
- **Diffusion phase**:
  - State update function applied to internal state 320 times
  - Register feedback bits and effective key bit j^t used in update
  - No keystream output BUT z^t used in both LFSR and NLFSR updates

Keytream generation

- z^t is used as keystream - not used in LFSR or NLFSR feedback

Slid Pairs for Simplified Plantlet

Consider component J (formed by combining K and C)

- J is autonomous
- Output of J is binary sequence, could be produced by cyclic register of length 80
  - Slid pairs can only arise from effective keys which are cyclic shifts of each other
- Fixed format of loaded state does not necessarily imply minimum phase shift
  - Since state update function is different in initialisation and keystream generation modes
- What size phase shifts are possible? Investigate in experiments

Experimental Investigation - Finding Slid Pairs

Experimental trials

- Select a Key and IV: (J_i, V_j)
- Initialise Simplified Plantlet and begin to produce keystream
- At each iteration of keystream generation, consider internal state as candidate initial state
  - Perform initialisation state update function in reverse, 320 times, and check if obtained state has format required for loaded state
- If so, note details of corresponding key, IV and phase shift (it's a slid pair!)
- Continue producing keystream and checking candidate states until 204,800 bits of keystream have been produced

Experimental Investigation - Finding Slid Pairs

Experiment Details

- Experiment performed with 12 different keys, 12 different IVs
  - Some patterned strings: 000...00; 01010101...01, etc
  - Some non-patterned strings
- Each key was used with each IV $\Rightarrow$ 144 trials were conducted
- In each trial (producing 204,800 bits of keystream), we recorded
  - the number of slid pairs occuring
  - the slid pair values (J_i, V_j), and
  - the size of the phase shifts

Simplified Plantlet Experiment Results

Number of slid pairs found per trial from initial pair (J_i, V_j)

	ExK_1	ExK_2	ExK_3	ExK_4	ExK_5	ExK_6	...	ExK_{12}
ExV_1	102	101	94	91	88	78	...	97
ExV_2	98	78	94	93	89	78	...	100
ExV_3	91	99	94	98	90	97	...	93
ExV_4	101	97	108	103	124	90	...	107
ExV_5	98	110	89	112	103	91	...	94
ExV_6	99	93	97	92	88	104	...	129
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
ExV_{10}	80	107	93	82	109	99	...	109
ExV_{11}	93	83	105	106	104	100	...	95
ExV_{12}	89	100	99	96	92	91	...	112

Simplified Plantlet Experiment Results

Size of Phase Shifts (Minimum, Mean, Maximum)

	ExK_1	ExK_2	ExK_3	ExK_4	ExK_5	...	ExK_{12}
ExV_1	17 1,989 12,441	4 2,026 12,832	14 2,168 9,176	6 2,239 13,114	62 2,315 14,250	... ...	13 2,102 9,805
ExV_2	70 2,086 9,510	73 2,627 10,578	5 2,141 13,402	14 2,156 9,433	5 2,245 14,408	... ...	9 2,038 7,676
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
ExV_{12}	72 2,223 15,927	7 2,009 13,302	26 2,042 9,118	1 2,121 9,096	21 2,197 10,862	... ...	1 1,774 8,679

Slid Pairs for Simplified Plantlet

Main findings

Slid pairs found in all trials

- Minimum number found: 72 (for ExK_7, ExV_1)
- Maximum number found: 129 (for ExK_{12}, ExV_6)
- Inspection of keys in slid pairs revealed
  - All keys in slid pairs are cyclic shifts of key used
  - Occurrence approximates proportion of distinct keys possible

Phase shift for slid pairs

- Minimum phase shift < 10 in over 30% of trials
- Mean phase shift approx 2000

Slid Pairs for Plantlet

Relationship between Simplified Plantlet and Plantlet Keys

- For a given IV, the sequence produced by Plantlet with key K can be produced by Simplified Plantlet with key $J = K \oplus C_4$

Phase Shifts and Slid Pairs

- Suppose for Simplified Plantlet, (J_0, V_0) produces a keystream, and (J_1, V_1) produces an α shifted keystream
- Consider the Plantlet keys that produce these sequences
  - Clearly, $K_0 = J_0 \oplus C_4$
  - Similarly, if $\alpha \bmod 80 = 0$, then $K_1 = J_1 \oplus C_4 = K_0$
  - If $\alpha \bmod 80 \neq 0$, then obtaining K_1 involves correcting for out-of-phase counter sequence:
    $K_1 = J_1 \oplus C_4 \oplus C_4 \lll \alpha \bmod 80$

Slid Pairs for Plantlet

- Suppose Plantlet is used with key K_0 to produce keystream
- The keys in slid pairs with phase shift α are of the form $(K_0 \lll (\alpha \bmod 80)) \oplus (C_4 \lll (\alpha \bmod 80)) \oplus C_4$
- To verify the Slid Pairs key relationship, experiments were performed following the process used for Simplified Plantlet
  - Slid Pairs found in all experimental trials
  - Minimum number found: 69 (for ExK_9, ExV_6)
  - Maximum number found: 122 (for ExK_1, ExV_6)
  - Phase shifts of less than 10 occurred in 28% of trials

Discussion

- For both Simplified Plantlet and Plantlet, slid pairs occurred for all (Key, IV) pairs used our experiments
  - Average phase shift approx. 2000, can be as small as 1
- For Simplified Plantlet, keys in slid pairs are cyclic shifts
  - Implication for patterned keys if period of key sequence < 80
  - Example: $01010101\ldots01$
- For Plantlet, relationship between keys in slid pairs slightly more complex
  - Combination of a cyclic shift of the key with a masking value obtained from the counter sequence
  - $(K_0 \lll (\alpha \bmod 80)) \oplus (C_4 \lll (\alpha \bmod 80)) \oplus C_4$
  - Since counter sequence has period 80, there are 79 effective masking values

Conclusion

- Where multiple keystreams will be produced from different (Key,IV) inputs, Plantlet keystreams are not all distinct and unpredictable - a relationship has been established
- Use the relationship between keys giving rise to slid pairs to divide the keyspace
  - Form sets of keys that can produce shifted keystreams
  - Size of each set is at most 80
- May be able to exploit this in TMD attacks - future work

Questions?