

Heraclitus

A LFSR based Stream Cipher with Key
Dependent Structure

Bernard Colbert
Deakin University

bernard.colbert@hotmail.com

Anthony H. Dekker
University of Ballarat

dekker@acm.org

Lynn M. Batten
Deakin University
(presenter)

lmbatten@deakin.edu.au

Introduction

- Heraclitus is a proof of concept cipher
- The aim is to demonstrate the existence of a cipher with the following characteristics
 - a key dependent structure
 - a large number of instances
 - each instance being cryptographically strong
- Based on A5.1 cipher
- Note that many choices of elements of a cipher are chosen from a larger set

A5.1

- A5.1 was designed in 1980s to secure air interface of mobile phones
- Has been analysed extensively
 - Golić , 1997
 - Biryukov, Shamir, and Wagner, 2000
 - Englund, Johansson, and Turan, Indocrypt 2007
 - Nohl and Krißler, August 2009
- Attacks are feasible due to small size of cipher
 - Allows for precomputation and tables
 - Structure was not compromised

Underlying Structure of A5.1

- Elements of A5.1 are
 - registers – number of lengths
 - the lengths are all co prime
 - feedback polynomials
 - irregular clocking mechanism
 - Note that no separate structure or element to introduce non linearity, entirely provided by clocking mechanism
- Each of these can be varied
- Heraclitus is based on A5.1 – however, these elements are dynamically chosen at cipher establishment

Overview of Cipher Instantiation

- The idea of Heraclitus is to use a 128 bit index χ to determine the structure of the cipher
- Note that this is different to the cipher key
- The construction of a instances is as follows:
 1. Determine the number of registers
 2. Determine length of each register
 3. Determine the feedback polynomial for each register
 4. The clocking mechanism is majority clocking as in A5.1
 5. Load the cipher key

Example (64 bit index)

01001001 10010110 00000010 11010010 01001001 10010110 00000010 11010010

01001001100 **10110000000** **10110100100**
10010011001 **01100000001** **01101001** **0**

5 registers (instead of 7) register lengths {25, 29, 32, 41, 43}

Polynomials:

$$x^{25} + x^{15} + x^{14} + x^{11} + x^9 + x^8 + x^3 + x^2 + 1$$

$$x^{29} + x^{19} + x^{15} + x^{10} + x^9 + x^8 + x^7 + x^6 + x^3 + x + 1$$

$$x^{32} + x^{22} + x^{15} + x^{13} + x^{12} + x^{10} + x^9 + x^8 + x^7 + x^2 + 1$$

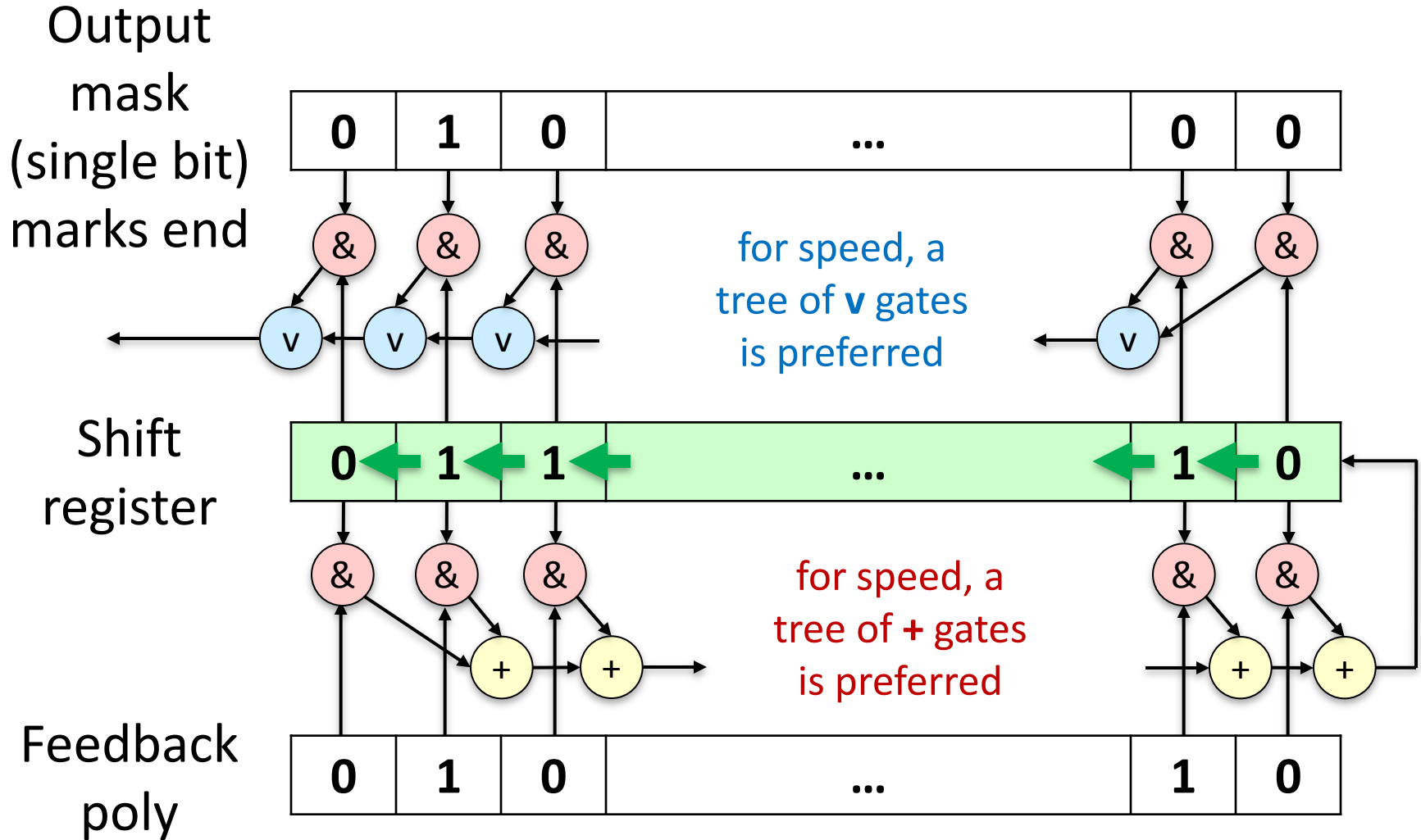
$$x^{41} + x^{31} + x^{15} + x^{14} + x^{13} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

$$x^{43} + x^{33} + x^{14} + x^{13} + x^7 + x^6 + x^4 + x^3 + 1$$

Registers

- Registers in Heraclitus vary
 - in number: {5, 7, 9, 11}
 - in length: {17, 19, 23, 25, 27, 29, 31, 32, 37, 41, 43, 47, 49, 53, 59, 61}
- Feedback polynomials are all primitive polynomials

Variable-length registers in hardware



Security

- Guessing registers is not feasible due to increased length of registers
- Golić Time Memory trade off is no longer feasible
- Refinements of Golić's attack are not feasible either
- Pre computation of tables and other characteristics is no longer feasible:
 - Increased size of each cipher
 - Large number of instances

Implementation

- Variations of Heraclitus has been implemented
 - Heraclitus 64
 - 64 bit index
 - Number of registers: 5 or 7
 - Lengths from {23, 25, 27, 29, 31, 32, 37, 41, 43, 47, 49}
 - Feedback polynomials stored in a table
 - 8 or 9 bits to specify registers & 11 or 8 bits per polynomial
 - A5+
 - Undergraduate cryptography project to implement
 - 128 bit index
 - Fixed number of registers
 - Length and polynomials vary
 - Feedback polynomials stored in a table

Efficiency

Variant	Setup time	Memory Runtime	Memory Storage
Heraclitus	$c \cdot 3 \times 10^6$ instructions	< 10kB	< 10kB
Heraclitus 64	8 table lookups	< 200 kB	2 MB
A5+	7 table lookups	1 MB	8 MB

Conclusions and Observations

- Demonstrated a secure and efficient cipher with varying structure exists and can be implemented in software
- Not yet implemented in hardware
- Each instance of Heraclitus has a secure structure