Security assessment of WhibOx 2017 candidates

Alexander Treff

WhibOx 2019 workshop

WhibOx 2017 rules

- Submit white-boxed AES-128 implementation
- Pure C code (no includes, libraries, ...)
- Source $\leq$ 50MB, binary $\leq$ 20 MB, avg. runtime $\leq$ 1s
- Designer goal: remain unbroken
- Attacker goal: break as fast as possible

WhibOx 2017 results

- 94 implementations (1 invalid)
- 13 earned > 0 points
  - 10 designers
- ALL broken
- Detailed presentation & write-up for winning challenge
- No (public) write-ups for others(?)

- How many can be broken in an *automated* way?
  - i.e., without (much) reverse engineering
  - DCA/DFA
- Classification by size, speed, security

Attack classification

- Automated
  - DCA
  - DFA
  - Higher-order DCA
- Manual effort
  - DCA after modification
    - Devirtualization
    - Removal of time-consuming code
  - DFA after modification
    - Removal of duplicate rounds
    - Removal of pseudo-randomness
  - Other methods
- Unbroken
  - Reverse engineering effort

Toolchain

- customized Intel PIN plugin for trace acquisition
- Jlsca for efficient DCA
- DFA script from SideChannelMarvels
- customized *aes-brute-force* tool for round-10-key-bruteforcing
  - *Hulk* from SideChannelMarvels does the same, but better

Differential computation analysis (DCA)

- Software counterpart to differential power analysis (DPA)
- Collect software execution traces
- Use statistical methods to figure out correct key bytes

DCA countermeasures

- Masking sensitive values throughout computation
  - Generate pseudo-randomness from input
- Artificially enlarging traces by dummy operations
  - Useful for contest
  - Not desired for real-world implementations

Differential fault analysis (DFA)

- Obtain correct output for specific input
- Induce faulty outputs by e.g. flipping bits
- Collect faulty outputs
- Compare against correct output
- Compute last round key
- Compute actual AES key

DFA countermeasures

- Compute results twice & compare
- Mute output/set output to unrelated
- Other countermeasures
  - <4 faulty state bytes in r9
  - >4 faulty state bytes in r9
  - Faulty state bytes in wrong position
  - "Skip" r9 vulnerability

Higher-order DCA

- Combine k samples of computation trace
- Exponential complexity
- Second-order is feasible for *some* implementations
- Reveals key for at least one first-order-resistant implementation
  - Apparently also breakable using other methods

DCA

- A lot of dummy-submissions
- Some using dual ciphers
  - Use different selection function (Klemsa model)
- ≥ 1 Challenge resistant against input-DCA but vuln. to output-DCA
- Majority breakable by DCA
  - 14 AES reference implementations
  - 19 T6_256 by *chaes*
  - 17 other
  - $=$ 50 in total
- Some more can be broken when modified

DFA

- Only applied when DCA failed
- Good against virtualized implementations (Tigress)
- Manual injection sometimes better than automated
- 3/4 columns sufficient (last can be brute-forced)
- Another 7 broken using script, 7 broken manually

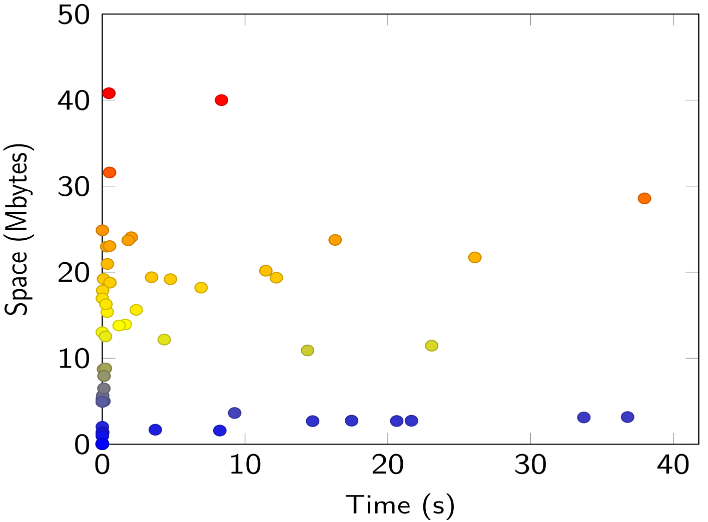
Modifications

- DCA improvements
  - Removal of dummy code
  - Removal of non-constant code
  - Removal of trace-enlarging code (JH hash computation)
  - Devirtualization
  - Breaks 5 more challenges using DCA
- DFA improvements
  - Removal of duplicate rounds
  - Removal of DFA protection (obviously)

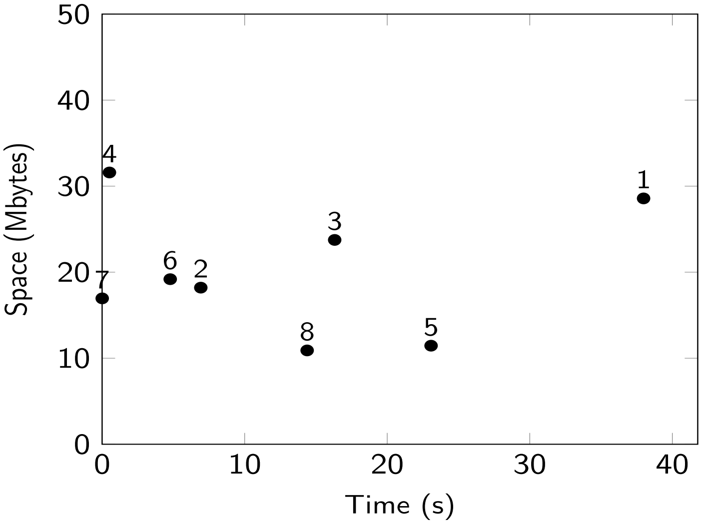
Leftover challenges

- All with > 0 points except *festive_jennings* (using DFA, 3/4 cols)
- Second generation by *kluxc3qa1* (5 submissions)
- Two more submissions by different authors
  - One of them encoding input bits as (0=0x00000000, 1=0xfffffff7), adding rk-bits as uint32, then bitsliced implementation
  - The other one using multiple nested lookups

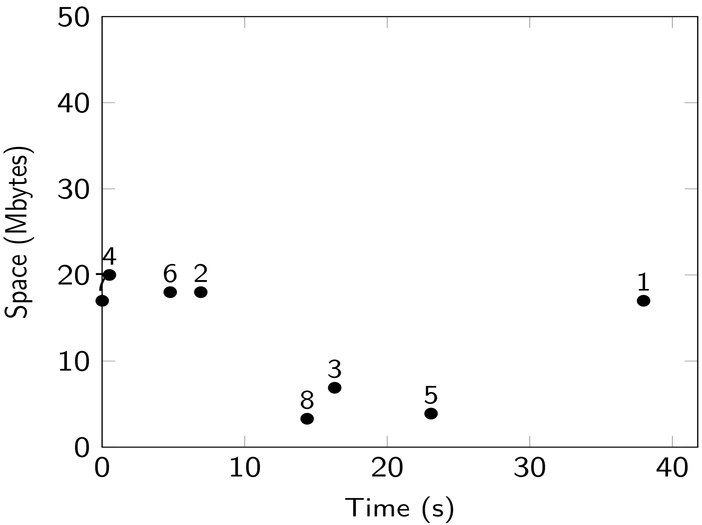
All source size



Top 8 source size



Top 8 binary size



Security considerations

- Unbroken = good, broken = bad?
- Can we combine: small, fast, secure?
  - Industry: "secure" $\approx$ not broken in $\leq x$ days
  - Only breakable with manual effort
  - Refresh implementation before it is broken
- Automated attacks vs. manual effort
- Break one, break all?