Optimisation heuristics in randomness testing

Karel Kubíček

2017-06-22

Cryptoprimitives and their cryptanalysis

- AES
  - 3 years competition
  - used for 17+ years
- DES
  - used for 24+ years

Cryptoprimitives and their cryptanalysis

- AES
  - 3 years competition
  - used for 17+ years
- DES
  - used for 24+ years
- Manual cryptanalysis

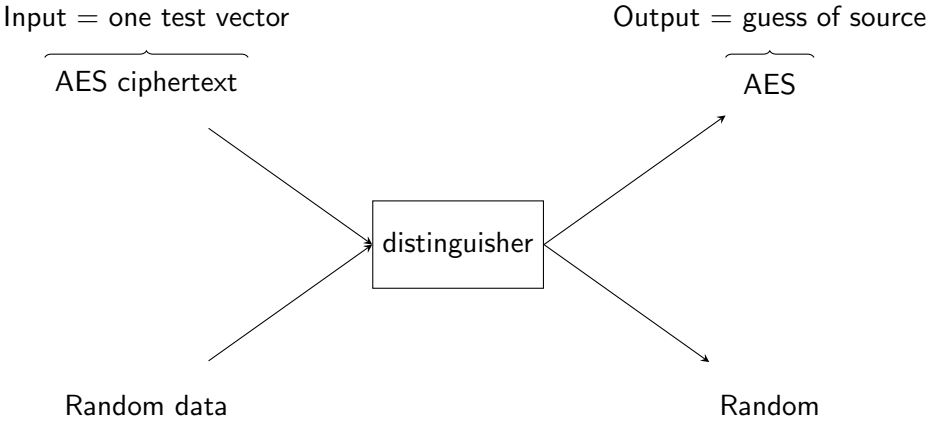
Cryptoprimitives and their cryptanalysis

- AES
  - 3 years competition
  - used for 17+ years
- DES
  - used for 24+ years
- Manual cryptanalysis
- Automation
  - statistical batteries: fixed
  - EACirc: adaptive to the data

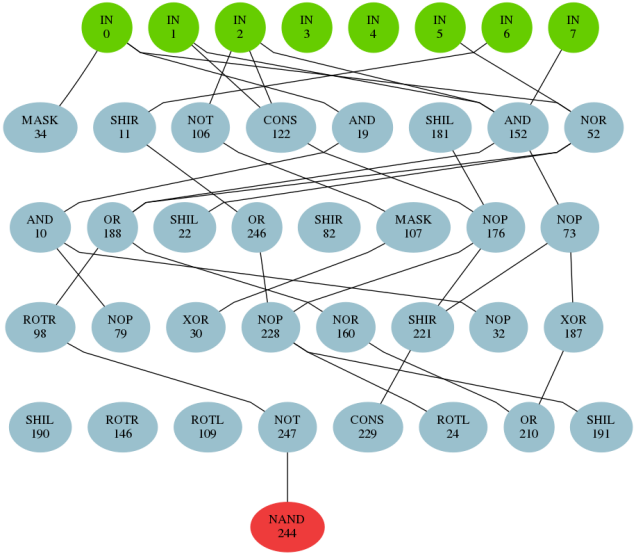
Goals of the work

- Study metaheuristics
  - analyse their application for randomness testing
- Try 4-5 metaheuristics
- Compare them using dataset of well-known crypto functions
  - Implement the dataset functions

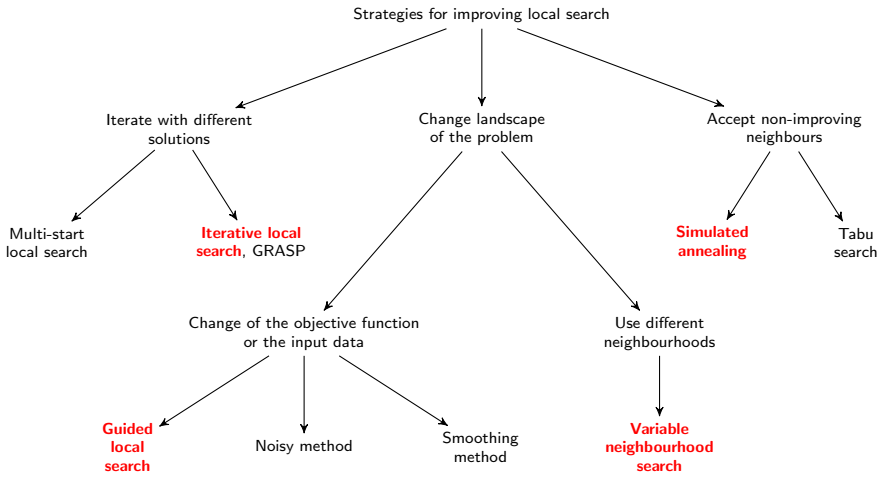
EACirc basic scheme



EACirc circuit – the distinguisher from previous slide



Metaheuristics classification



Evaluation dataset

- Well-known functions for benchmarking
  - SHA-3 finalists: Keccak, BLAKE, Grøstl, JH, Skein
  - eSTREAM winners: HC-128, Rabbit, Salsa20, SOSEMANUK, Grain
  - other: AES, DES, TEA, RC4

Evaluation dataset

- Well-known functions for benchmarking
  - SHA-3 finalists: Keccak, BLAKE, Grøstl, JH, Skein
  - eSTREAM winners: HC-128, Rabbit, Salsa20, SOSEMANUK, Grain
  - other: AES, DES, TEA, RC4
- Tool for data generation
  - Data are highly customisable
  - Thesis data published online (and used by other research teams)

Metaheuristic: Iterative local search – baseline

Function\rounds	0	1	2	3	4	5	6	7	8	9	10
rnd_rnd	0.01112	–	–	–	–	–	–	–	–	–	–
AES	–	1.0	1.0	0.160	0.015	–	–	–	–	–	–
BLAKE	1.0	0.110	0.007	0.012	–	–	–	–	–	–	–
Grain	–	–	1.0	0.006	0.007	–	–	–	–	–	–
Grøstl	–	–	1.0	0.013	0.013	0.013	–	–	–	–	–
HC-128	0.009	0.007	–	–	–	–	–	–	–	–	–
JH	–	–	–	–	–	–	1.0	0.015	0.012	–	–
Keccak	–	1.0	1.0	0.018	0.017	–	–	–	–	–	–
MD6	–	–	–	–	–	–	–	–	0.774	0.009	0.007
Rabbit	0.014	0.009	–	–	–	–	–	–	–	–	–
RC4	–	0.01	–	–	–	–	–	–	–	–	–
Salsa20	–	–	1.0	0.016	0.01	–	–	–	–	–	–
SINGLE-DES	–	–	–	1.0	0.204	0.017	0.010	–	–	–	–
Skein	0.012	–	–	–	–	–	–	–	–	–	–
SOSEMANUK	0.012	0.010	–	–	–	–	–	–	–	–	–
TEA	–	–	–	1.0	0.444	0.009	0.010	–	–	–	–
TRIPLE-DES	–	–	1.0	0.010	0.015	0.012	–	–	–	–	–

Metaheuristic: Simulated annealing

Function\rounds	0	1	2	3	4	5	6	7	8	9	10
rnd_rnd	0.01681	–	–	–	–	–	–	–	–	–	–
AES	–	1.0	1.0	0.305	0.013	–	–	–	–	–	–
BLAKE	1.0	0.051	0.015	0.017	–	–	–	–	–	–	–
Grain	–	–	1.0	0.019	0.011	–	–	–	–	–	–
Grøstl	–	–	1.0	0.012	0.019	0.014	–	–	–	–	–
HC-128	0.009	0.015	–	–	–	–	–	–	–	–	–
JH	–	–	–	–	–	–	1.0	0.018	0.02	–	–
Keccak	–	1.0	1.0	0.012	0.013	–	–	–	–	–	–
MD6	–	–	–	–	–	–	–	–	0.419	0.019	0.015
Rabbit	0.024	0.016	–	–	–	–	–	–	–	–	–
RC4	–	0.011	–	–	–	–	–	–	–	–	–
Salsa20	–	–	1.0	0.013	0.021	–	–	–	–	–	–
SINGLE-DES	–	–	–	1.0	0.093	0.019	0.021	–	–	–	–
Skein	0.014	–	–	–	–	–	–	–	–	–	–
SOSEMANUK	0.013	0.022	–	–	–	–	–	–	–	–	–
TEA	–	–	–	1.0	0.244	0.013	0.015	–	–	–	–
TRIPLE-DES	–	–	1.0	0.015	0.018	0.023	–	–	–	–	–

Metaheuristic: Guided local search

Function\rounds	0	1	2	3	4	5	6	7	8	9	10
rnd_rnd	0.01110	–	–	–	–	–	–	–	–	–	–
AES	–	1.0	1.0	0.164	0.005	–	–	–	–	–	–
BLAKE	1.0	0.260	0.012	0.012	–	–	–	–	–	–	–
Grain	–	–	1.0	0.010	0.013	–	–	–	–	–	–
Grøstl	–	–	1.0	0.015	0.016	0.014	–	–	–	–	–
HC-128	0.021	0.012	–	–	–	–	–	–	–	–	–
JH	–	–	–	–	–	–	1.0	0.011	0.003	–	–
Keccak	–	1.0	1.0	0.021	0.016	–	–	–	–	–	–
MD6	–	–	–	–	–	–	–	–	0.995	0.020	0.006
Rabbit	0.006	0.015	–	–	–	–	–	–	–	–	–
RC4	–	0.011	–	–	–	–	–	–	–	–	–
Salsa20	–	–	1.0	0.007	0.011	–	–	–	–	–	–
SINGLE-DES	–	–	–	1.0	0.496	0.014	0.013	–	–	–	–
Skein	0.009	–	–	–	–	–	–	–	–	–	–
SOSEMANUK	0.010	0.007	–	–	–	–	–	–	–	–	–
TEA	–	–	–	1.0	0.729	0.011	0.008	–	–	–	–
TRIPLE-DES	–	–	1.0	0.008	0.012	0.009	–	–	–	–	–

Metaheuristic: Variable neighbourhood search

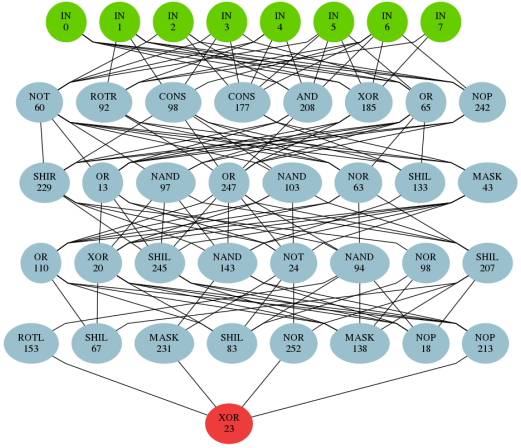
Function\rounds	0	1	2	3	4	5	6	7	8	9	10
rnd_rnd	0.01150	–	–	–	–	–	–	–	–	–	–
AES	–	1.0	1.0	0.182	0.009	–	–	–	–	–	–
BLAKE	1.0	0.157	0.006	0.012	–	–	–	–	–	–	–
Grain	–	–	1.0	0.010	0.011	–	–	–	–	–	–
Grøstl	–	–	1.0	0.014	0.007	0.017	–	–	–	–	–
HC-128	0.015	0.015	–	–	–	–	–	–	–	–	–
JH	–	–	–	–	–	–	1.0	0.005	0.010	–	–
Keccak	–	1.0	1.0	0.018	0.014	–	–	–	–	–	–
MD6	–	–	–	–	–	–	–	–	0.992	0.006	0.010
Rabbit	0.017	0.010	–	–	–	–	–	–	–	–	–
RC4	–	0.004	–	–	–	–	–	–	–	–	–
Salsa20	–	–	1.0	0.007	0.012	–	–	–	–	–	–
SINGLE-DES	–	–	–	1.0	0.441	0.006	0.016	–	–	–	–
Skein	0.006	–	–	–	–	–	–	–	–	–	–
SOSEMANUK	0.013	0.012	–	–	–	–	–	–	–	–	–
TEA	–	–	–	1.0	0.877	0.009	0.013	–	–	–	–
TRIPLE-DES	–	–	1.0	0.013	0.007	0.017	–	–	–	–	–

Metaheuristics comparison

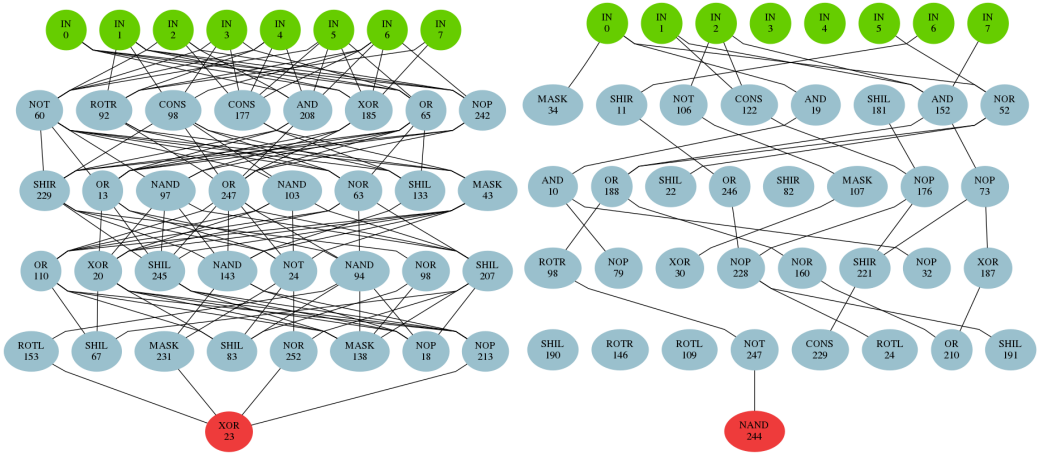
$\textbf{Function}_{\textbf{rounds}} \backslash \textbf{Metaheuristic}$	Iterated local search	Simulated annealing	Guided local search	Variable neigh. search
AES_3	0.160	**0.305**	0.164	0.182
$BLAKE_1$	0.110	0.051	**0.260**	0.157
$MD6_8$	0.774	0.419	**0.995**	0.992
Single DES_4	0.204	0.093	**0.496**	0.441
TEA_4	0.444	0.244	0.729	**0.877**

- Differences are small
- Guided local search seems to be the best
  - because it has no performance loss and:

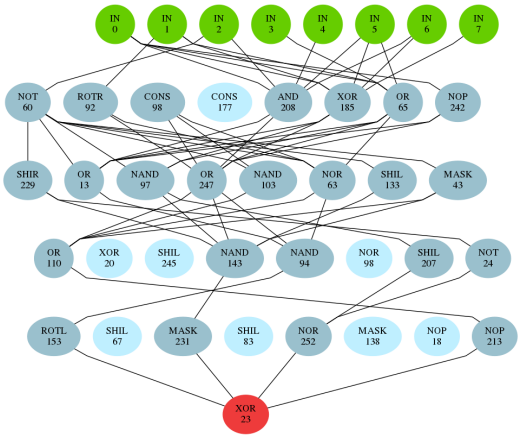
Cryptanalysis from the circuit



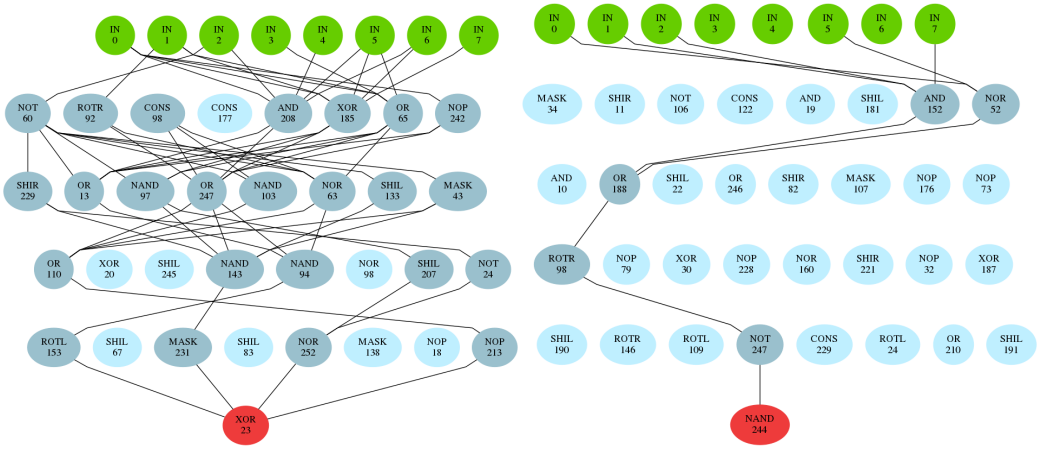
Cryptanalysis from the circuit



Cryptanalysis from the circuit – pruning



Cryptanalysis from the circuit – pruning



Comparison with other tools

- EACirc competitors = statistical batteries
  - NIST STS – worse or same (to EACirc)
  - Dieharder – similar or slightly better
  - TestU01 – far better

Conclusion

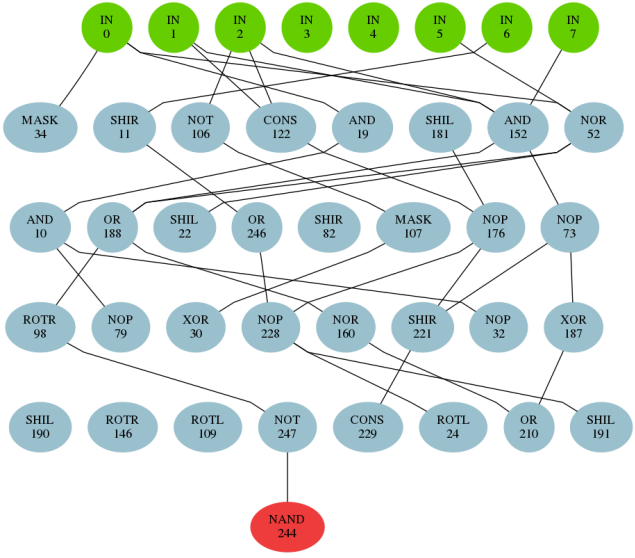
- Various metaheuristics explored, three implemented in EACirc
  - evaluated on many well-known cryptographic functions
- Guided local search used in EACirc
  - mainly due to circuits analysis
- Metaheuristics additionally tested on
  - evolutionary algorithms on polynomial representation
  - neural networks for randomness testing
  - ongoing cooperation with Ca' Foscari University of Venice and BUT
- Developed standalone tool for crypto data generation
  - planed as publication on NordSec

Conclusion

- Various metaheuristics explored, three implemented in EACirc
  - evaluated on many well-known cryptographic functions
- Guided local search used in EACirc
  - mainly due to circuits analysis
- Metaheuristics additionally tested on
  - evolutionary algorithms on polynomial representation
  - neural networks for randomness testing
  - ongoing cooperation with Ca' Foscari University of Venice and BUT
- Developed standalone tool for crypto data generation
  - planed as publication on NordSec
- Thank you for your attention, questions?

"Connection bug" explanation

"Connection bug" explanation



Workaround

- Remove 0 p-values
- Change circuit if p-value $= 0$
- Substitute 0 p-values with random (0,1)
- Completely new evaluation
- Metaheuristics

Baseline random-random experiment

Function\rounds	0	1	2	3	4	5	6	7	8	9	10
rnd_rnd	**0.01112**	–	–	–	–	–	–	–	–	–	–
AES	–	1.0	1.0	0.160	0.015	–	–	–	–	–	–
BLAKE	1.0	0.110	0.007	0.012	–	–	–	–	–	–	–
Grain	–	–	1.0	0.006	0.007	–	–	–	–	–	–
Grøstl	–	–	1.0	0.013	0.013	0.013	–	–	–	–	–
HC-128	0.009	0.007	–	–	–	–	–	–	–	–	–
JH	–	–	–	–	–	–	1.0	0.015	0.012	–	–
Keccak	–	1.0	1.0	0.018	0.017	–	–	–	–	–	–
MD6	–	–	–	–	–	–	–	–	0.774	0.009	0.007
Rabbit	0.014	0.009	–	–	–	–	–	–	–	–	–
RC4	–	0.01	–	–	–	–	–	–	–	–	–
Salsa20	–	–	1.0	0.016	0.01	–	–	–	–	–	–
SINGLE-DES	–	–	–	1.0	0.204	0.017	0.010	–	–	–	–
Skein	0.012	–	–	–	–	–	–	–	–	–	–
SOSEMANUK	0.012	0.010	–	–	–	–	–	–	–	–	–
TEA	–	–	–	1.0	0.444	0.009	0.010	–	–	–	–
TRIPLE-DES	–	–	1.0	0.010	0.015	0.012	–	–	–	–	–

Metaheuristics comparison

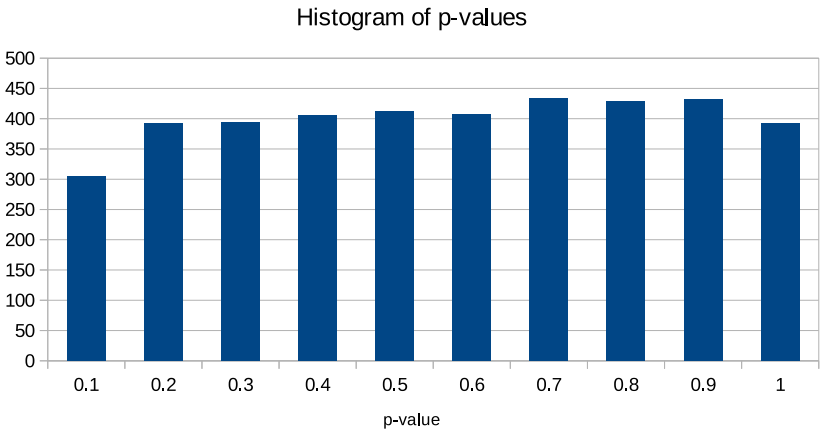
- Remove 0 p-values $\rightarrow$ 0.01681
- Change circuit if p-value $= 0 \rightarrow$ cycling
- Substitute 0 p-values with random (0,1) $\rightarrow$ 0.01267
- Completely new evaluation $\rightarrow$ weaker
- Metaheuristics

Metaheuristics comparison

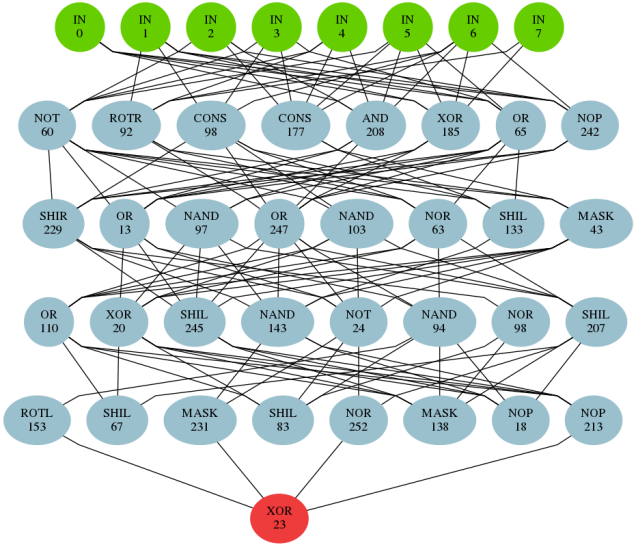
- Remove 0 p-values $\rightarrow$ 0.01681
- Change circuit if p-value $= 0 \rightarrow$ cycling
- Substitute 0 p-values with random (0,1) $\rightarrow$ 0.01267
- Completely new evaluation $\rightarrow$ weaker
- Metaheuristics

Function\Metaheuristic	**Iterated local search**	**Simulated annealing**	**Guided local search**	**Variable neigh. search**
rnd_rnd	0.01112	0.01681	0.01110	0.01150

"Substitute 0 *p*-values with random (0,1)" analysis



Guided local search – dense circuits



Guided local search – sparse circuits