Cryptographically Strong S-boxes Generated by Modified Immune Algorithm

Georgi Ivanov, IMI-BAS, Bulgaria
Nikolay Nikolov, IMI-BAS, Bulgaria
Svetla Nikova, KU Leuven, Belgium

BalkanCryptSec, Koper, Slovenia
$3^{rd} - 4^{th}$ September, 2015

Preliminaries
- S-box Construction Techniques
- S-box Target Set of Cryptographic Criteria
- Known Best Results Obtained

Goal and Basic Idea
- Main Goal and Basic Idea
- Artificial Immune Systems

Special Immune Algorithm
- Algorithm Description
- Experimental Results

Conclusion

Bibliography

Preliminaries
- S-box Construction Techniques
- S-box Target Set of Cryptographic Criteria
- Known Best Results Obtained

Goal and Basic Idea
- Main Goal and Basic Idea
- Artificial Immune Systems

Special Immune Algorithm
- Algorithm Description
- Experimental Results

Conclusion

Bibliography

S-box Construction Techniques

- Pseudo-random Generation
- Algebraic Constructions
- Heuristic Approaches

Pseudo-random Generation

- S-box entries are generated from a table of random numbers.
- Test the S-box for suitability with respect to a target set of cryptographic criteria.
- Unproductive as the size n of the input space increases:
  - Trade-off between cryptographic criteria.
  - Small number of good S-boxes among all in the search space.

Algebraic Constructions

- Rely on proven mathematical principles. The AES S-box [5].
- Popular as S-boxes obtained are known to be optimal with respect to most cryptographic criteria.
- Typically, do not produce a great number of S-boxes.
- Simple algebraic structure. Affine equivalent.
- Potential source for future security concerns if any of these S-boxes appears to be vulnerable to algebraic attacks [4].

Heuristic Approaches

- Based on evolutionary searching.
- A few S-boxes are iteratively improved with respect to one or more cryptographic properties until:
  - some reasonable number of iterations or execution time is reached.
  - some chosen in advance specific threshold values for these properties are achieved.
- Produce a great number of non-optimal S-boxes.
- Include: Hill Climbing Method [13, 14], Simulated Annealing Method [3], Genetic Algorithms [8, 19], etc.

S-box Target Set of Cryptographic Criteria

- Nonlinearity
- Minimal Algebraic Degree
- Differential Uniformity
- Autocorrelation

Nonlinearity

For better resistance to *Linear Cryptanalysis* [12], the *nonlinearity* N_S of an $(n \times n)$ *bijective* S-box S should be maximized.

Nonlinearity

For better resistance to *Linear Cryptanalysis* [12], the *nonlinearity* N_S of an $(n \times n)$ *bijective* S-box S should be maximized.

The largest non-trivial value of S-box *Linear Approximation Table*, denoted by LAT_S, should be minimized.

Nonlinearity

For better resistance to *Linear Cryptanalysis* [12], the *nonlinearity* N_S of an $(n \times n)$ *bijective* S-box S should be maximized.

$$\Updownarrow$$

The largest non-trivial value of S-box *Linear Approximation Table*, denoted by LAT_S, should be minimized.

$$\Updownarrow$$

The minimal *nonlinearity* among all *nonlinearities* of *component* Boolean functions of the S-box should be maximized. That is:

$$N_S = \min_{v \in \mathbb{B}^n \setminus \{0\}} N_{vS} \longrightarrow max.$$

Minimal Algebraic Degree

For better resistance to *Linear Cryptanalysis* [12], *Low Order Approximation Attacks* [15], *Higher-order Differential Attacks* [11], *Interpolation Attack* [9] and *Algebraic Attacks* [4], the *minimal algebraic degree* of an $(n \times n)$ *bijective* S-box S should be high.

Minimal Algebraic Degree

For better resistance to *Linear Cryptanalysis* [12], *Low Order Approximation Attacks* [15], *Higher-order Differential Attacks* [11], *Interpolation Attack* [9] and *Algebraic Attacks* [4], the *minimal algebraic degree* of an $(n \times n)$ *bijective* S-box S should be high.

$$\Updownarrow$$

The minimal *algebraic degree* among all the *algebraic degrees* of *component* Boolean functions of the S-box should be maximized:

$$deg(S) = \min_{v \in \mathbb{B}^n \setminus \{0\}} deg(vS) \longrightarrow max, \text{ where}$$

where $deg(vS)$ is the number of variables of the largest product term of ANF_{vS} having a non-zero coefficient.

Differential Uniformity

For better resistance to *Differential Cryptanalysis* [1], *differential uniformity* δ of an $(n \times n)$ *bijective* S-box S should be minimized.

Differential Uniformity

For better resistance to *Differential Cryptanalysis* [1], *differential uniformity* δ of an $(n \times n)$ *bijective* S-box S should be minimized.

$$\Updownarrow$$

The largest non-trivial value of the S-box *Difference Distribution Table*, denoted by DDT_S, should be minimized. In other words:

$$\delta = \max_{\alpha \in \mathbb{B}^n \setminus \{0\}} \max_{\beta \in \mathbb{B}^n} |\{x \in \mathbb{B}^n | S(x) \oplus S(x \oplus \alpha) = \beta\}| \longrightarrow min.$$

If $\delta = 2$, then the S-box is referred to as an APN S-box.
Existence of APN permutations for even $n > 6$ is an open problem.

Autocorrelation

In order to improve the *Avalanche Effect* [6] of the cipher, the largest non-trivial absolute *autocorrelation* value of of an $(n \times n)$ *bijective* S-box S, denoted by $AC(S)_{max}$, should be minimized.

Autocorrelation

In order to improve the *Avalanche Effect* [6] of the cipher, the largest non-trivial absolute *autocorrelation* value of of an $(n \times n)$ *bijective* S-box S, denoted by $AC(S)_{max}$, should be minimized.

$$\Updownarrow$$

The maximal non-trivial absolute *autocorrelation* value among all absolute *autocorrelation* values (*absolute indicators*) of the S-box *component* Boolean functions should be minimized. In other words:

$$AC(S)_{max} = \max_{v \in \mathbb{B}^n \setminus \{0\}} AC_{max}(vS) \longrightarrow min.$$

Known Best Results Obtained

Table: Best (8×8) *bijective* S-boxes generated

Generation methods/properties	N_S	$deg(S)$	$AC(S)_{max}$	δ
Pseudo-random Generation [13, 14]	98	-	-	-
Finite Field Inversion [16]	**112**	**7**	**32**	**4**
Hill Climbing Method [13]	100	-	-	-
Genetic Alg/Hill Climbing [14]	100	-	-	-
Simulated Annealing Method [3]	102	-	80	-
Special Genetic Algorithm [19]	104	-	-	-
Tweaking Method [7]	106	7	56	6
Gradient Descent Method [10]	104	7	80	8
4-uniform Perm. Method [17, 18]	98	-	-	4
Reversed Genetic Algorithm [8]	110	7	40	4
Reversed Genetic Algorithm [8]	112	7	32	6

Preliminaries
- S-box Construction Techniques
- S-box Target Set of Cryptographic Criteria
- Known Best Results Obtained

Goal and Basic Idea
- Main Goal and Basic Idea
- Artificial Immune Systems

Special Immune Algorithm
- Algorithm Description
- Experimental Results

Conclusion

Bibliography

Main Goal and Basic Idea

- Obtain by some heuristic a variety of (8×8) *bijective*, affine non-equivalent S-boxes with cryptographic properties close to those of the finite field inversion-based S-boxes and with far more complex algebraic structure.

- Hill Climbing Method [13] or Simulated Annealing Method [3] - productive but ineffective just by themselves.

- Genetic Algorithm
  - Classic Genetic Algorithm, based on the traditional "climbing-up" approach [19].

  - Special Genetic Algorithm, based on the reverse "skiing-down" approach - Reversed Genetic Algorithm [8].

- Try something different - why not an Immune Algorithm?

Artificial Immune Systems

(Clonal Selection Algorithms)

- Inspired by the process and mechanisms of the biological immune system.
- Work with only one candidate solution, corresponding to the most appropriate type of general immune cells (lymphocytes) that will fight a specific pathogen.
- Candidate solution - subject to:
  - *proliferation*.
  - *somatic hypermutation*.
  - *selection*.

Preliminaries
- S-box Construction Techniques
- S-box Target Set of Cryptographic Criteria
- Known Best Results Obtained

Goal and Basic Idea
- Main Goal and Basic Idea
- Artificial Immune Systems

Special Immune Algorithm
- Algorithm Description
- Experimental Results

Conclusion

Bibliography

Algorithm Description

STEP 1 (Initialization)

- Define an integer n.
- Generate a random $(n \times n)$ *bijective* S-box S_0.

Algorithm Description

STEP 1 (Initialization)

- Define an integer n.
- Generate a random $(n \times n)$ *bijective* S-box S_0.

STEP 2 (Initial selection)

- Start the *MHCM* with S_0 as an input.
- Obtain $S = MHCM(S_0)$ - the $(n \times n)$ *bijective* S-box of the lowest cost: $cost(S) \longrightarrow min$

Algorithm Description

STEP 1 (Initialization)

- Define an integer n.
- Generate a random $(n \times n)$ *bijective* S-box S_0.

STEP 2 (Initial selection)

- Start the *MHCM* with S_0 as an input.
- Obtain $S = MHCM(S_0)$ - the $(n \times n)$ *bijective* S-box of the lowest cost: $cost(S) \longrightarrow min$

STEP 3 (Somatic hypermutation)

- Twice apply $mutation_1(\cdot)$ with S as an input: $S_1 = mutation_1(S)$ and $S_2 = mutation_1(S)$.
- Twice apply $mutation_2(\cdot)$ with S as an input: $S_3 = mutation_2(S)$ and $S_4 = mutation_2(S)$.
- Obtain four different $(n \times n)$ *bijective* S-boxes S_1, S_2, S_3 and S_4.

Algorithm Description

STEP 4 (Selection)

- Start *MHCM* with each of S_1, S_2, S_3 and S_4.
- Obtain low-cost S-boxes S_1', S_2', S_3' and S_4': $S_1' = MHCM(S_1)$, $S_2' = MHCM(S_2)$, $S_3' = MHCM(S_3)$ and $S_4' = MHCM(S_4)$.
- Compare the costs of S_1', S_2', S_3' and S_4', and set S' to be the S-box of the lowest cost.

Algorithm Description

STEP 4 (Selection)

- Start *MHCM* with each of S_1, S_2, S_3 and S_4.
- Obtain low-cost S-boxes S_1', S_2', S_3' and S_4': $S_1' = MHCM(S_1)$, $S_2' = MHCM(S_2)$, $S_3' = MHCM(S_3)$ and $S_4' = MHCM(S_4)$.
- Compare the costs of S_1', S_2', S_3' and S_4', and set S' to be the S-box of the lowest cost.

STEP 5 (Stopping criterion)

- If some chosen in advance threshold number of iterations or execution time is reached, STOP.
- Otherwise, set S to S' and go to step 3.

Algorithm Description

- $cost(S) = cost_1(S).cost_2(S).cost_3(S)$, where

Algorithm Description

- $cost(S) = cost_1(S).cost_2(S).cost_3(S)$, where
  - $cost_1(S) = \sum_{c<d \in \mathbb{B}^n \setminus \{0\}} \sum_{\omega \in \mathbb{B}^n} \mid |\widehat{F}_{cS}(\omega)|^3 - |\widehat{F}_{dS}(\omega)|^3 \mid^7$.
  - $cost_2(S) = \sum_{c \in \mathbb{B}^n \setminus \{0\}} \sum_{\omega \in \mathbb{B}^n} |\widehat{F}_{cS}(\omega) - 21|^7$, [3, 19].
  - $cost_3(S) = \sum_{\delta_{11} \neq \delta_{ij} \in DDT_S} (\delta_{ij} - 1)^2.(\delta_{ij} - 2)^2.(\delta_{ij} - 4)^2$.

Algorithm Description

- $cost(S) = cost_1(S).cost_2(S).cost_3(S)$, where
  - $cost_1(S) = \sum_{c<d\in\mathbb{B}^n\setminus\{0\}} \sum_{\omega\in\mathbb{B}^n} \mid |\widehat{F}_{cS}(\omega)|^3 - |\widehat{F}_{dS}(\omega)|^3 \mid^7$.
  - $cost_2(S) = \sum_{c\in\mathbb{B}^n\setminus\{0\}} \sum_{\omega\in\mathbb{B}^n} |\widehat{F}_{cS}(\omega) - 21|^7$, [3, 19].
  - $cost_3(S) = \sum_{\delta_{11}\neq\delta_{ij}\in DDT_S} (\delta_{ij} - 1)^2.(\delta_{ij} - 2)^2.(\delta_{ij} - 4)^2$.
- $mutation_1(S)$ - swap 2 neighbouring elements of S at positions $p-1$ and p, where $p \in [2, 2^n]$ is chosen at random.
- $mutation_2(S)$ - swap q neighbouring elements of S at position p, where q is chosen at random in $[2, 8]$ and p is chosen at random accordingly.

Experimental Results ($n = 8$)

Generation methods/properties	N_S	$deg(S)$	$AC(S)_{max}$	δ
Pseudo-random Generation [13, 14]	98	-	-	-
Finite Field Inversion [16]	**112**	**7**	**32**	**4**
Hill Climbing Method [13]	100	-	-	-
Genetic Alg/Hill Climbing [14]	100	-	-	-
Simulated Annealing Method [3]	102	-	80	-
Special Genetic Algorithm [19]	104	-	-	-
Tweaking Method [7]	106	7	56	6
Gradient Descent Method [10]	104	7	80	8
4-uniform Perm. Method [17, 18]	98	-	-	4
Reversed Genetic Algorithm [8]	110	7	40	4
Reversed Genetic Algorithm [8]	112	7	32	6
Special Immune Algorithm	104	7	88	6

Experimental Results ($n = 8$)

Generation methods/properties	N_S	$deg(S)$	$AC(S)_{max}$	δ
~~Pseudo-random Generation~~ [13, 14]	~~98~~	-	-	-
Finite Field Inversion [16]	**112**	**7**	**32**	**4**
~~Hill Climbing Method~~ [13]	~~100~~	-	-	-
~~Genetic Alg/Hill Climbing~~ [14]	~~100~~	-	-	-
~~Simulated Annealing Method~~ [3]	~~102~~	-	~~80~~	-
~~Special Genetic Algorithm~~ [19]	~~104~~	-	-	-
Tweaking Method [7]	**106**	**7**	**56**	**6**
~~Gradient Descent Method~~ [10]	~~104~~	~~7~~	~~80~~	~~8~~
~~4-uniform Perm. Method~~ [17, 18]	~~98~~	-	-	~~4~~
Reversed Genetic Algorithm [8]	**110**	**7**	**40**	**4**
Reversed Genetic Algorithm [8]	**112**	**7**	**32**	**6**
Special Immune Algorithm	**104**	**7**	**88**	**6**

Preliminaries
- S-box Construction Techniques
- S-box Target Set of Cryptographic Criteria
- Known Best Results Obtained

Goal and Basic Idea
- Main Goal and Basic Idea
- Artificial Immune Systems

Special Immune Algorithm
- Algorithm Description
- Experimental Results

Conclusion

Bibliography

Conclusion

The presented special immune algorithm:

- is a new heuristic "climbing-up" approach for bijective S-box generation.
- is able to produce large sets of affine inequivalent bijective S-boxes of high nonlinearity and low differential uniformity.
- has succeeded to narrow the distance to the finite field inversion-based S-boxes with respect to nonlinearity and differential uniformity.
- will provide an alternative in case some new, more powerful, algebraic attacks against the AES-type of S-boxes appear.

Finite field inversion-based S-boxes remain the optimal found with respect to the target set of criteria but heuristics are catching up.

Future work

The work provided can be extended in several future directions:

- Apply some changes in the number of the *mutation* functions or in the functions themselves so as (8×8) bijective S-boxes with $N > 104$ and $\delta = 4$ to be produced. At least, from [8], we know that such S-boxes exist.

- Apply some changes in the *cost* function or in the *mutation* functions so as (6×6) *APN permutations*, non-equivalent to the one in [2], to be searched for.

- Apply some changes in the *cost* or *mutation* functions so as (8×8) bijective S-boxes with $N > 112$ or $\delta = 2$ to be searched for (open problems).

Thank you!

Preliminaries
- S-box Construction Techniques
- S-box Target Set of Cryptographic Criteria
- Known Best Results Obtained

Goal and Basic Idea
- Main Goal and Basic Idea
- Artificial Immune Systems

Special Immune Algorithm
- Algorithm Description
- Experimental Results

Conclusion

Bibliography

E. Biham and A. Shamir.

Differential cryptanalysis of des-like cryptosystems.

In *Advances in Cryptology CRYPTO90*, volume 537 of *LNCS*, pages 2–21. Springer Verlag, 1991.

K. Browning, J. Dillon, M. McQuistan, and A. Wolfe.

An apn permutation in dimension six.

Finite Fields: Theory and Applications, Contemporary Mathematics, 518:33–42, 2010.

J.A. Clark, J.L. Jacob, and S. Stepney.

The design of s-boxes by simulated annealing.

New Generation Computing Archive, 23(3), September 2005.

N. T. Courtois and J. Pieprzyk.

Cryptanalysis of block ciphers with overdefined systems of equations.

In *Advances in Cryptology - ASIACRYPT02*, volume 2501 of *LNCS*, pages 267–287. Springer Verlag, 2002.

J. Daeman and V. Rijmen.

The design of Rijndael: AES The advanced Encryption Standard.

Springer Verlag, 2002.

H. Feistel.

Cryptography and computer privacy.

Scientific American, 228(5):15–23, 1973.

J. Fuller and W. Millan.

Linear redundancy in s-boxes.

In *FSE'03*, volume 2887 of *LNCS*, pages 74–86. Springer, 2003.

G. Ivanov, N. Nikolov, and S. Nikova.

Reversed genetic algorithms for generation of bijective s-boxes with good cryptographic properties.

IACR Cryptology ePrint Archive (2014), Report 2014/801, http://eprint.iacr.org/2014/801.pdf.

T. Jakobsen and L. Knudsen.

The interpolation attack on block ciphers.

In *FSE'97*, volume 1267, pages 28–40. Springer Verlag, 1997.

O. Kazymyrov, V. Kazymyrova, and R. Oliynykov.

A method for generation of high-nonlinear s-boxes based on gradient descent.

IACR Cryptology ePrint Archive (2013).

L. R. Knudsen.

Truncated and higher order differentials.

In *FSE94*, volume 1008 of *LNCS*, pages 196–211. Springer Verlag, 1995.

M. Matsui.

Linear cryptanalysis method for des cipher.

In *Advances in Cryptology EUROCRYPT93*, volume 765 of *LNCS*, pages 386–397. Springer Verlag, 1994.

W. Millan.

How to improve the nonlinearity of bijective s-boxes.

In *Australian Conference on Information Security and Privacy 1998*, volume 1438, pages 181–192. Springer Verlag, 1998.

W. Millan, L. Burnett, G. Carter, A. Clark, and E. Dawson.

Evolutionary heuristics for finding cryptographically strong s-boxes.

In *ICICS99*, volume 1726 of *LNCS*, pages 263–274. Springer, 1999.

W. L. Millan.

Low order approximation of cipher functions.

In *Cryptography: Policy and Algorithms Conference, Proceedings*, volume 1029 of *LNCS*, pages 144–155. Springer Verlag, 1996.

K. Nyberg.

Differentially uniform mappings for cryptography.

In *Advances in Cryptology EUROCRYPT93*, volume 765 of *LNCS*, pages 55–64. Springer Verlag, 1994.

L. Qu, Y. Tan, C. Li, and G. Gong.

More constructions of differentially 4-uniform permutations on $\mathbb{F}_{2^{2k}}$.

In *arxiv.org/pdf/1309.7423*, 2013.

L. Qu, Y. Tan, C. Tan, and C. Li.

Constructing differentially 4-uniform permutations over $\mathbb{F}_{2^{2k}}$ via the switching method.

IEEE Transactions on Inform. Theory, 59(7):4675–4686, 2013.

P. Tesar.

A new method for generating high non-linearity s-boxes.

Radioengineering, 19(1):23–26, 2010.