About

BSbox-tools: Software for representation, defining and computing the most important cryptographic properties of Boolean and Vector Boolean functions (S-boxes)

Current release: v0.2

The problems related to representation, defining and computing the most important cryptographic properties of Boolean and Vector Boolean functions (S-boxes) require effective algorithms. With the increasing amount of input data, the problem requires more computational resources. To compute some of the cryptographic properties (linearity, autocorrelation, algebraic degree, differential uniformity) very effective algorithms have to be realized. These algorithms are suitable for parallel implementation.

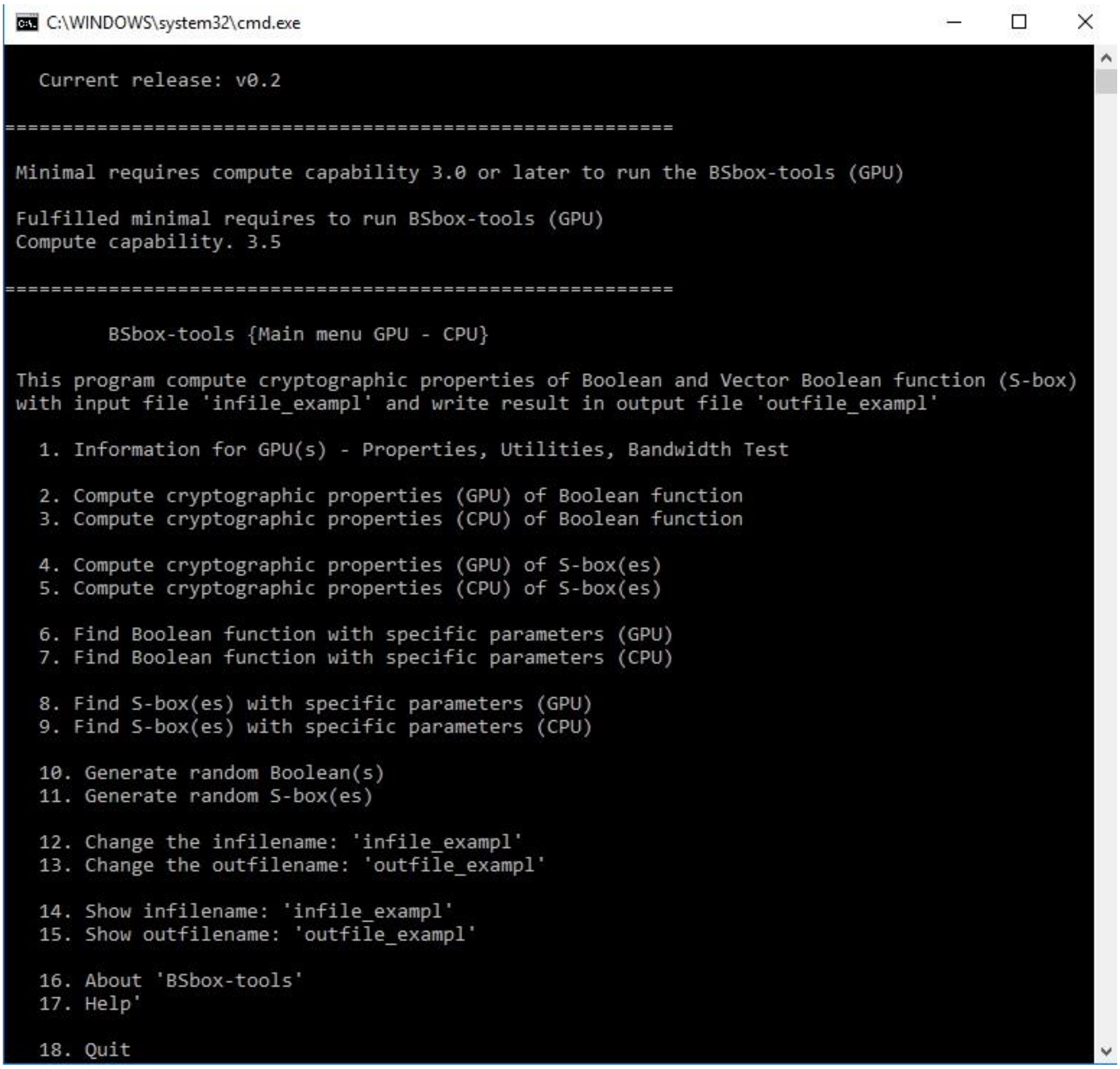
What is BSbox-tools?

BSbox-tools is developed software for representation, defining and computing the most important cryptographic properties of Boolean and Vector Boolean functions (S-boxes). For development of this software is use our CUDA oriented library BoolSPLG (Boolean S-box Properties Library for GPUs), a library with parallel algorithms for Boolean functions and S-boxes for GPU. To use this software there is no need knowledge of a programming language. BoolSPL (Boolean S-box parallel library for GPU) provides, reusable software components for every layer of the CUDA programming model [5]. BoolSPLG is a library consisting procedures for analysis and compute cryptographic properties of Boolean and Vector Boolean function (S-box). Our procedures have function for auto grid configuration. Most of the functions are designed to compute the data in registers because they offer the highest bandwidth.

Overview of BSbox-tools

BSbox-tools is specialized console application program for representation, defining and computing the most important cryptographic properties of Boolean and Vector Boolean functions (S-boxes). With other words BSbox-tools represents a console interface program on the BoolSPL library. The version BSbox-tools_v0.2 is based on BoolSPL_v0.2 library version. This application combines CPU and GPU functionality.

On figure 1 is show main menu of BSbox-tools console interface program. How we can see it is given some information about the program as current version, minimal requires compute capability, notification if minimal requires are fulfilled. After initial information their info about input and output files and then follows menu with the options.

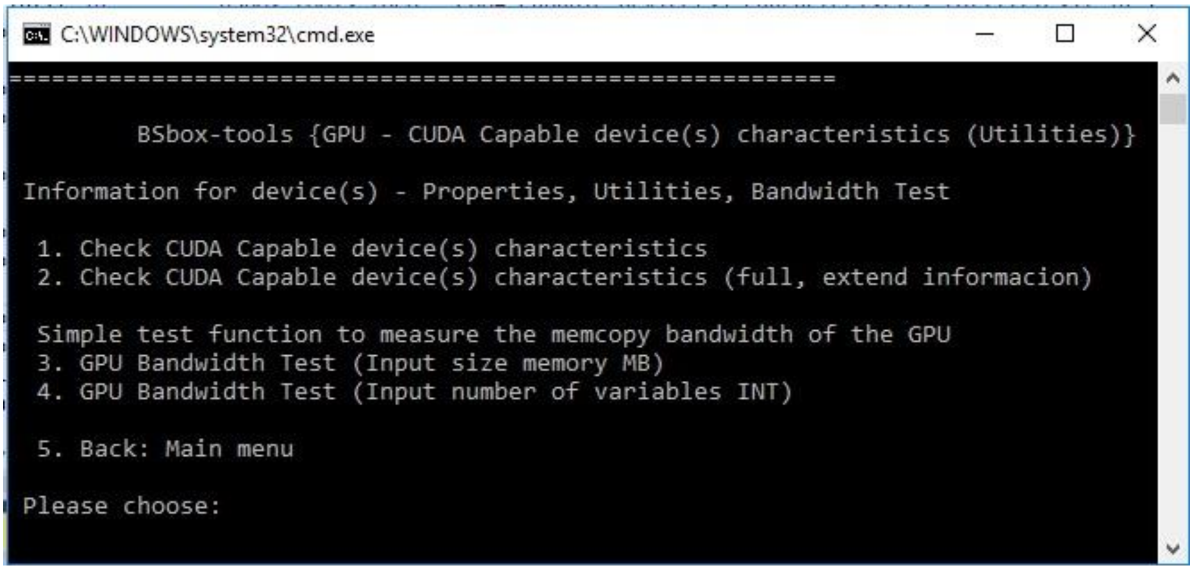


C:\WINDOWS\system32\cmd.exe

```
   Current release: v0.2

==========================================================

 Minimal requires compute capability 3.0 or later to run the BSbox-tools (GPU)

 Fulfilled minimal requires to run BSbox-tools (GPU)
 Compute capability. 3.5

==========================================================

        BSbox-tools {Main menu GPU - CPU}

 This program compute cryptographic properties of Boolean and Vector Boolean function (S-box)
 with input file 'infile_exampl' and write result in output file 'outfile_exampl'

   1. Information for GPU(s) - Properties, Utilities, Bandwidth Test

   2. Compute cryptographic properties (GPU) of Boolean function
   3. Compute cryptographic properties (CPU) of Boolean function

   4. Compute cryptographic properties (GPU) of S-box(es)
   5. Compute cryptographic properties (CPU) of S-box(es)

   6. Find Boolean function with specific parameters (GPU)
   7. Find Boolean function with specific parameters (CPU)

   8. Find S-box(es) with specific parameters (GPU)
   9. Find S-box(es) with specific parameters (CPU)

   10. Generate random Boolean(s)
   11. Generate random S-box(es)

   12. Change the infilename: 'infile_exampl'
   13. Change the outfilename: 'outfile_exampl'

   14. Show infilename: 'infile_exampl'
   15. Show outfilename: 'outfile_exampl'

   16. About 'BSbox-tools'
   17. Help'

   18. Quit
```

Figure 1. Main menu of "BSbox-tools {Main menu GPU - CPU}" console interface program

The first submenu (figure 2) - “1. Information for GPU(s) - Properties, Utilities, Bandwidth Test” can help to check GPU device Properties, Utilities and to perform Bandwidth Test.

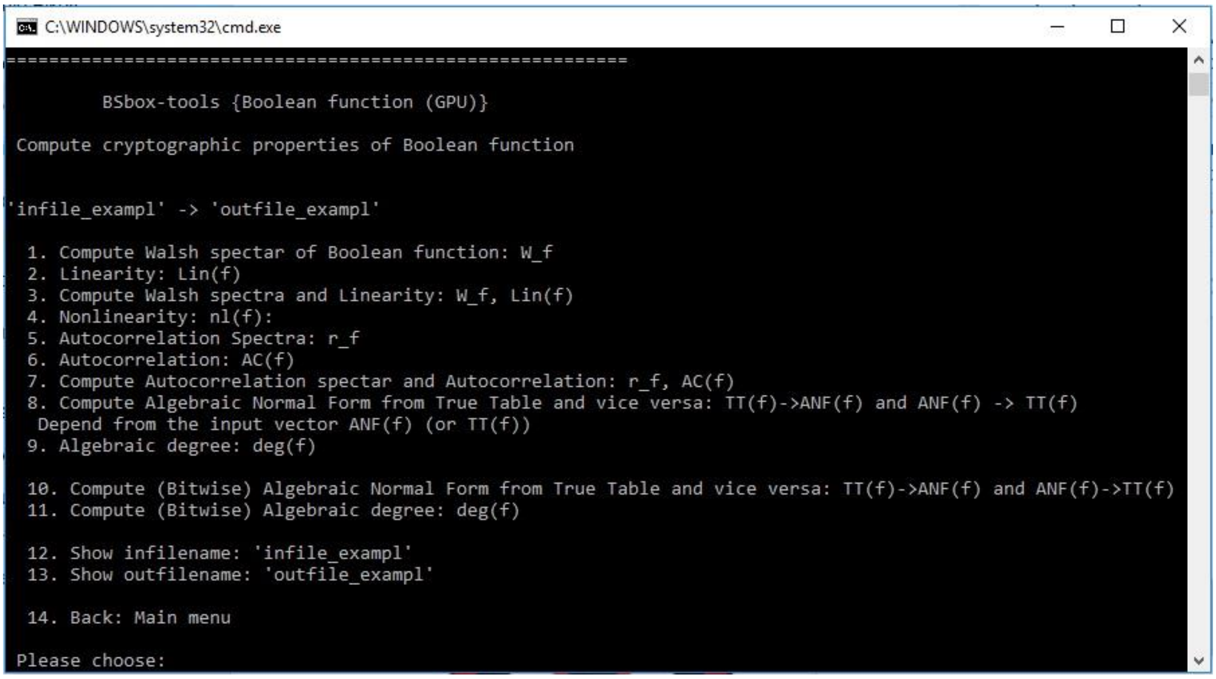


```
==========================================================

        BSbox-tools {GPU - CUDA Capable device(s) characteristics (Utilities)}

 Information for device(s) - Properties, Utilities, Bandwidth Test

  1. Check CUDA Capable device(s) characteristics
  2. Check CUDA Capable device(s) characteristics (full, extend informacion)

  Simple test function to measure the memcopy bandwidth of the GPU
  3. GPU Bandwidth Test (Input size memory MB)
  4. GPU Bandwidth Test (Input number of variables INT)

  5. Back: Main menu

 Please choose:
```

Figure 2. Submenu "1. Information for GPU(s) - Properties, Utilities, Bandwidth Test"

The second submenu (figure 3) from the main menu "2. Compute cryptographic properties (GPU) of Boolean function" have option that compute cryptographic properties Walsh spectra, Linearity, Nonlinearity, Autocorrelation Spectra, Autocorrelation, Algebraic degree, Algebraic Normal Form from True Table and vice versa of Boolean function. GPU in brackets means that these procedures are executed parallel on GPU.

The third submenu from the main menu is the same as second, but the procedures are intended for CPU execution.

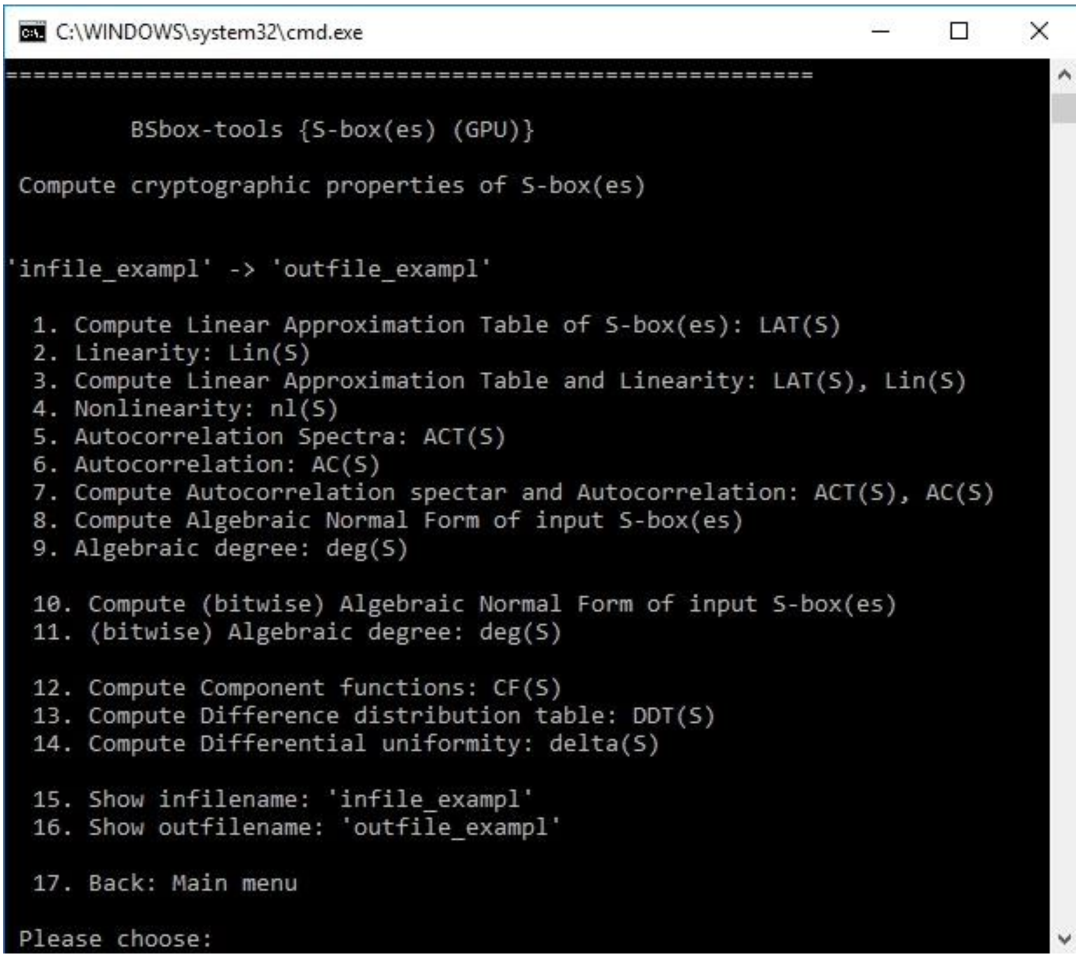


```
==========================================================

        BSbox-tools {Boolean function (GPU)}

 Compute cryptographic properties of Boolean function


'infile_exampl' -> 'outfile_exampl'

  1. Compute Walsh spectar of Boolean function: W_f
  2. Linearity: Lin(f)
  3. Compute Walsh spectra and Linearity: W_f, Lin(f)
  4. Nonlinearity: nl(f):
  5. Autocorrelation Spectra: r_f
  6. Autocorrelation: AC(f)
  7. Compute Autocorrelation spectar and Autocorrelation: r_f, AC(f)
  8. Compute Algebraic Normal Form from True Table and vice versa: TT(f)->ANF(f) and ANF(f) -> TT(f)
   Depend from the input vector ANF(f) (or TT(f))
  9. Algebraic degree: deg(f)

  10. Compute (Bitwise) Algebraic Normal Form from True Table and vice versa: TT(f)->ANF(f) and ANF(f)->TT(f)
  11. Compute (Bitwise) Algebraic degree: deg(f)

  12. Show infilename: 'infile_exampl'
  13. Show outfilename: 'outfile_exampl'

  14. Back: Main menu

 Please choose:
```

Figure 3. Submenu "2. Compute cryptographic properties (GPU) of Boolean function"

The fourth submenu (figure 4) from the main menu “4. Compute cryptographic properties (GPU) of S-box(es)” have option that compute cryptographic properties Linear Approximation Table, Linearity, Nonlinearity, Autocorrelation Spectra, Autocorrelation, (bitwise) Algebraic Normal Form of input S-box(es), (bitwise) Algebraic degree, Component functions, Difference distribution table, Differential uniformity. GPU in brackets means that these procedures are executed parallel on GPU.

The fifth submenu from the main menu is the same as fourth, but the procedures are intended for CPU execution.



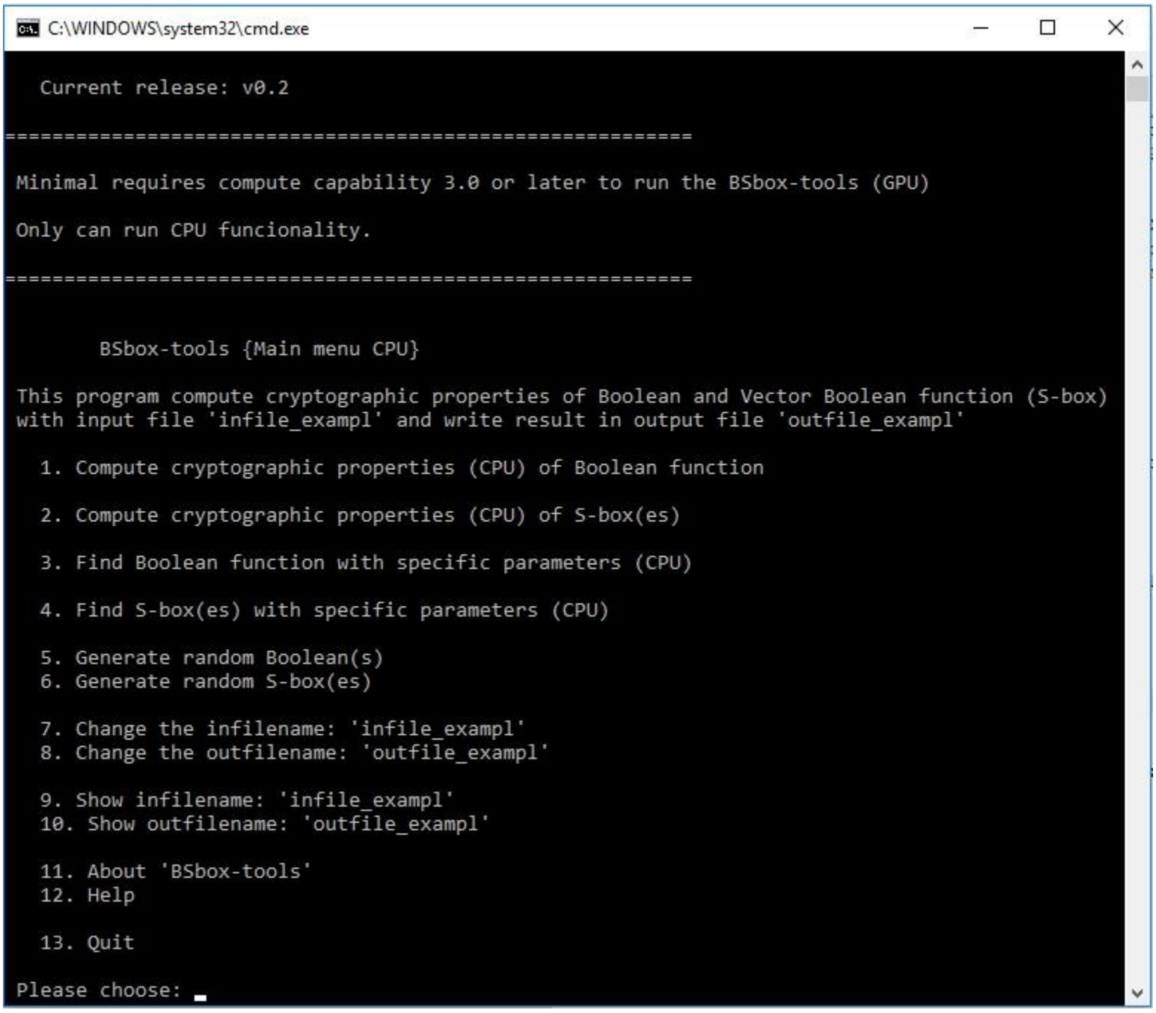
```
C:\WINDOWS\system32\cmd.exe
=========================================================

        BSbox-tools {S-box(es) (GPU)}

 Compute cryptographic properties of S-box(es)


'infile_exampl' -> 'outfile_exampl'

  1. Compute Linear Approximation Table of S-box(es): LAT(S)
  2. Linearity: Lin(S)
  3. Compute Linear Approximation Table and Linearity: LAT(S), Lin(S)
  4. Nonlinearity: nl(S)
  5. Autocorrelation Spectra: ACT(S)
  6. Autocorrelation: AC(S)
  7. Compute Autocorrelation spectar and Autocorrelation: ACT(S), AC(S)
  8. Compute Algebraic Normal Form of input S-box(es)
  9. Algebraic degree: deg(S)

  10. Compute (bitwise) Algebraic Normal Form of input S-box(es)
  11. (bitwise) Algebraic degree: deg(S)

  12. Compute Component functions: CF(S)
  13. Compute Difference distribution table: DDT(S)
  14. Compute Differential uniformity: delta(S)

  15. Show infilename: 'infile_exampl'
  16. Show outfilename: 'outfile_exampl'

  17. Back: Main menu

 Please choose:
```

Figure 4. Submenu “4. Compute cryptographic properties (GPU) of S-box(es)

Sixth and seventh submenu from the main menu are similar as second and third but here we can specify input parameters for investigation of Boolean functions. Eighth and ninth submenu from the main menu are similar as fourth and fifth but here we can specify input parameters for investigation of S-boxes. Option ten form the main menu give interface for random generation of test Boolean(s) function. Input parameter after choice option ten are input number of bits (n) and number for generate Boolean(s) functions. The next option (option eleven) from the main menu give interface for random generation of test S-box(es). Input parameter after choice option eleven are input number of bits (n) and number for generate S-box(es). Whit twelfth and thirteen form the main menu can be change the name of input or output file.

Option fourteen and fifteen form the main menu can show content of input or output file. There is also option for information as “About” and “Help”.

In order to gain full functionality from BSbox-tools there is minimal requires for Nvidia GPU that need to be with Compute capability bigger then 3. If hardware doesn’t fulfil minimal requires it is possible to use BSbox-tools but with only CPU functionality figure 5.

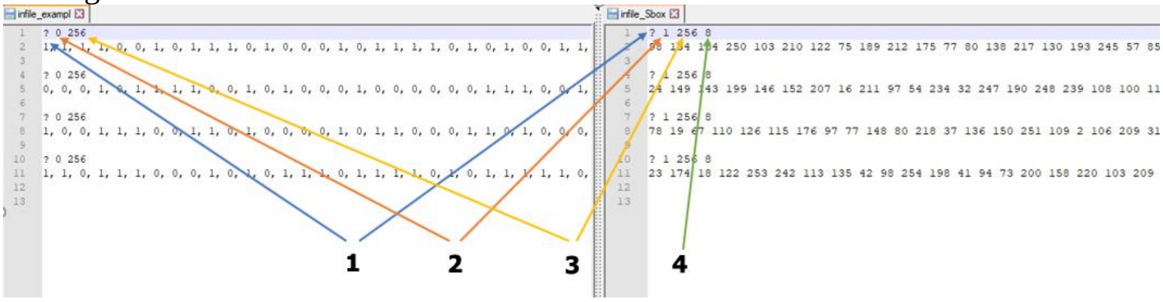


```
  Current release: v0.2

==========================================================

Minimal requires compute capability 3.0 or later to run the BSbox-tools (GPU)

Only can run CPU funcionality.

==========================================================


       BSbox-tools {Main menu CPU}

This program compute cryptographic properties of Boolean and Vector Boolean function (S-box)
with input file 'infile_exampl' and write result in output file 'outfile_exampl'

  1. Compute cryptographic properties (CPU) of Boolean function

  2. Compute cryptographic properties (CPU) of S-box(es)

  3. Find Boolean function with specific parameters (CPU)

  4. Find S-box(es) with specific parameters (CPU)

  5. Generate random Boolean(s)
  6. Generate random S-box(es)

  7. Change the infilename: 'infile_exampl'
  8. Change the outfilename: 'outfile_exampl'

  9. Show infilename: 'infile_exampl'
  10. Show outfilename: 'outfile_exampl'

  11. About 'BSbox-tools'
  12. Help

  13. Quit

Please choose: _
```

Figure 5. Main menu of "BSbox-tools {Main menu CPU}" console interface program

Input file "infile_exampl" format and output file (outfile_exampl)

Input file format have few input parameters that is similar for input Boolean function and S-box figure 6.



```
? 0 256
1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1,

? 0 256
0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1,

? 0 256
1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0,

? 0 256
1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0,
```

```
? 1 256 8
98 134 184 250 103 210 122 75 189 212 175 77 80 138 217 130 193 245 57 85

? 1 256 8
24 149 143 199 146 152 207 16 211 97 54 234 32 247 190 248 239 108 100 11

? 1 256 8
78 19 67 110 126 115 176 97 77 148 80 218 37 136 150 251 109 2 106 209 31

? 1 256 8
23 174 18 122 253 242 113 135 42 98 254 198 41 94 73 200 158 220 103 209
```

Figure 6. Input file (Boolean function, S-box)

Input parameters:

1. Character “?” indicate start of parameter section.
2. Character “0” indicate Boolean function, character “1” indicate S-box.
3. Size of Boolean function or S-box.
4. Number of bits (n).

Bits of Boolean function can set only with spaces.

Output file will contain result from computing cryptographic properties of Boolean function (or S-box).

How do I get started using BSbox-tools?

For the windows user there need to be have install Visual C++ Redistributable Packages for Visual Studio 2013 to be able to run and use the BSbox-tools.

There is possibility to rebuild the BSbox-tools from the source code. Have in mind that BSbox-tools is developed on Visual Studio 2013. In order to rebuild BSbox-tools you need to have BoolSPL library (v0.2) (BoolSPL is implemented as a C++ header library) and there is timer function that is use from Boost C++ Libraries.

Reference and Publications related to the BSbox-tools

[1] D. Bikov and I. Bouyukliev, BoolSPLG: A library with parallel algorithms for Boolean functions and S-boxes for GPU, preprint.
[2] D. Bikov, I. Bouyukliev, Parallel Fast Walsh Transform Algorithm and its implementation with CUDA on GPUs. Cybernetics and Information Technologies. Cybernetics and Information Technologies 18, 21–43 (2018).
[3] D. Bikov and I. Bouyukliev, Parallel Fast Mobius (Reed-Muller) Transform and its Implementation with CUDA on GPUs, Proceedings of PASCO 2017, Kaiserslautern, Germany, Germany — July 23 - 24, 2017, ISBN: 978-1-4503-5288-8 (improvement presented in this publication are implemented in current v0.2 BoolSPL library)
[4] D. Bikov and I. Bouyukliev, BoolSPLG: A library with parallel algorithms for Boolean functions and S-boxes for GPU, Poster session, PUMPS+AI 2018, Barcelona, Spain.
[5] CUDA homepage, Availaible on: https://developer.nvidia.com/cuda-zone

Additional - Reference and Publications related to the BSbox-tools

[1] I. Bouyukliev, D, Bikov, Applications of the binary representation of integers in algorithms for boolean functions, Proceedings of the Forty Fourth Spring Conference of the Union of Bulgarian Mathematicians SOK “Kamchia”, (2015), pp.161-166, ISSN: 1313-3330
[2] D. Bikov, I. Bouyukliev, Walsh Transform Algorithm and its Parallel Implementation with CUDA on GPUs, Proceedings of 25 YEARS FACULTY OF MATHEMATICS AND INFORMATICS, Veliko Tarnovo, Bulgaria, (2015), pp. 29-34, ISBN: 978-619-00-0419-6
[3] D. Bikov, I. Bouyukliev, A. Stojanova, Benefit of Using Shared Memory in Implementation of Parallel FWT Algorithm with CUDA C on GPUs, Proceedings of 7th International Conference Information Technologies and Education Development, Zrenjanin, Serbia, (2016) pp.250-256, ISBN 978-86-7672-285-3

[4] I. Bouyukliev, D. Bikov, S. Bouyuklieva, S-Boxes from Binary Quasi-Cyclic Codes, Electronic Notes in Discrete Mathematics Volume 57, (2017), pp. 67–72

[5] D. Bikov, I. Bouyukliev, S. Bouyuklieva, Bijective S-boxes of Different Sizes Obtained from Quasi-Cyclic Codes, submitted.