CryptoBinary Artifact Evaluation Documentation

Note: This documentation is designed for running each tool in a Linux environment.

1 Reproducible Only

1.1 Aligot

We were unable to run our benchmark with Aligot due to the incompatibility of the trace tool required by Aligot. However, we can still reproduce Aligot's original experiments.

1.1.1 Install the Tool

The original documentation for Aligot can be found at `https://github.com/j04n/aligot`. Aligot only works with `python2`, and it requires python libraries `pydot` and `networkx`.

1.1.2 Run the Tool

First, run the following command to enter the Aligot directory:

```
$ cd /path-to-aligot/aligot/vanilla/comparison
```

Then, run the following command to detect TEA in binary

```
$ python main.py examples/result_TEA.txt
```

You will see the detection result in terminal:

```
> Testing LDF 1 ...
> Heuristic :  Remove constants from parameters
> Comparison with TEA...

!!  Identification successful:  TEA decryption with:

==> Encrypted text (8 bytes) :  0x0123456789abcdef...

==> Key (16 bytes) :  0xdeadbee1deadbee2deadbee3deadbee4...

==> Decrypted text (8 bytes) :  0xdf5ec1536e089494...
```

This indicate that we can reproduce the Aligot evaluation with their own program. However, we cannot You can also run the following commands to test for RC5 and AES algorithm.

```
$ python main.py examples/result_AES_noasm_OD.txt
$ python main.py examples/result_result_MD5_noasm_OD.txt
```

2 Replication

2.1 DRACA

DRACA is a Windows binary tool that can only be run using Wine.

2.1.1 Install the Tool

DRACA is a tool that comes as a Windows binary only, and you can download from `http://www.literatecode.com/draca`. Compiling or dependencies are not needed.

2.1.2 Run the Tool

You can execute the following command to run DRACA with the selected binary program:

```
$ wine /path-to-draca/draca.exe path-of-binary
```

For example, if you want to test if DRACA can detect SHA-1 compiled by GCC with -O1 optimization flag, you will see the output in terminal like:

```
DRACA. Draft Crypto Analyzer.  Version 0.5.7b by Ilya O. Levin Preliminary detection and analysis
of crypto algorithms within executables.
File:  /home/artifacts/benchmarks/crypto_function/sha1/gcc/sha1_gcc_O1, 45040 byte(s)
analyzing...  done
results:  * SHA-1 - 100%
total 1 algorithm(s) recognized
```

We can see that DRACA has successfully detected SHA-1.

2.2 CryptoKnight

CryptoKnight is a machine learning-based tool that requires a training process before running cryptographic function detection.

2.2.1 Install the Tool

We have completed all pre-settings and installed the required packages on our server. To install CryptoKnight on other machine, you can download the source code from `https://github.com/AbertayMachineLearningGroup/CryptoKnight`. Then, you need to follow the instruction on their Github page to install the dependencies. Note that CryptoKnight requires python2. Besides that, you also need to install a python2 library `future`. Meanwhile, before using CryptoKnight to detect a cryptographic function, you also need to follow the instruction on their Github to setup the CryptoKnight. In our experience CryptoKnight is not easy to install in a new machine, so we recommend that you use our server to run CryptoKnight.

2.2.2 Run the Tool

First, you need to navigate to the CryptoKnight directory with the following command:

```
$ cd /path-to-CryptoKnight/
```

You can execute the following command to run the CryptoKnight with selected binary program.

```
$ python knight.py --evaluate path-of-binary
```

For example, if you want to test if CryptoKnight can detect AES compiled by GCC without optimization flag, you will see the output in terminal like:

```
[+] Full trace report:  /home/artifacteval/artifacts/tools/CryptoKnight/report.txt
[+] Feature Blocks:  1032
[+] Total Loop Count:  31
[+] Total Loop Iterations:  860

[+] End Time:  2024-08-27 15:27:08.851916
[+] Analysis Time:  00:00:00

[!]  Multiple primitives detected!

[+] Classification:
[*] ’Rivest-Shamir-Adleman (RSA)’
[*] ’Advanced Encryption Standard (AES)’
```

Note that you may need to wait a few seconds to see the analysis result.

2.3 Findcrypt2

2.3.1 Install the Tool

FindCrypt2 is an IDA Pro based tool. You can find the download link and instructions on how to load FindCrypt2 in IDA Pro at `https://www.aldeid.com/wiki/IDA-Pro/plugins/FindCrypt2`. You need to copy `findcrypt.plw` in the IDAProInstallDir/plugins directory.

2.3.2 Run the Tool

To run the tool, you need to open IDA Pro and load the binary program. Then click Edit, Plugins, FindCrypt2. You can skip this tool if you do not have IDA Pro.

2.4 HCD

HCD is a Windows binary tool that can only be run using Wine. It also requires X11 forwarding.

2.4.1 Install the Tool

HCD is a tool that comes as a Windows binary only, and you can download from `https://webscene.ir/tools/show/Hash-and-Crypto-Detector-1.4`. Compiling or dependencies are not needed.

2.4.2 Run the Tool

You can execute the following command to run the HCD with selected binary program.

```
$ wine /path-to-HCD/HCD.exe path-of-binary
```

Then, you will see the HCD GUI and the detection results. Note that HCD can only accept Windows binaries in `.exe` format.

2.5 PEiD KANAL

PEiD KANAL is a Windows binary tool that can only be run using Wine. It also requires X11 forwarding.

2.5.1 Install the Tool

PEiD KANAL is a tool that comes as a Windows binary only, and you can download from `http://www.dcs.fmph.uniba.sk/zri/6.prednaska/tools/PEiD/plugins/kanal.htm`. Compiling or dependencies are not needed.

2.5.2 Run the Tool

To run PEiD KANAL, you need to execute the following command.

```
$ wine /path-to-PEiD-KANAL/PEiD-0.95-KANAL/PEiD.exe
```

Then, you will see the PEiD KANAL GUI. First, click the button with three dots in the upper right corner to select the binary file. Another window will pop up where you can select our benchmark. Note that you need to choose "All files" in the "Files of type" drop-down list. After opening the binary, you will return to the PEiD KANAL main window. Click the right arrow button in the lower right corner, then select Plugins and Krypto ANALyzer. Finally, you will see the detection results.

2.6 SignSrch

SignSrch is a Windows binary tool that can only be run using Wine.

2.6.1 Install the Tool

SignSrch is a tool that comes as a Windows binary only, and you can download from `https://aluigi.altervista.org/mytoolz.htm`. Compiling or dependences are not needed.

2.6.2 Run the Tool

You can execute the following command to run the SignSrch with selected binary program.

```
$ wine tosignsrch/signsrch.exe path-of-binary]
```

For example, if you want to test if SignSrch can detect SHA-256 compiled by GCC with -O2 optimization flag, you will see output in the terminal like:

```
Signsrch 0.2.4
by Luigi Auriemma
e-mail:  aluigi@autistici.org
web:  aluigi.org
optimized search function by Andrew http://www.team5150.com/~andrew/
disassembler engine by Oleh Yuschuk

- open file "/home/artifacts/benchmarks/crypto_function/sha256/gcc/sha256_gcc_O2"
- 18040 bytes allocated
- load signatures
- open file Z:\home\artifacts\tools\signsrch\signsrch.sig
- 3075 signatures in the database
- start 4 threads
- start signatures scanning:

offset num description [bits.endian.size]
--------------------------------------------
00001896 876 SHA256 Initial hash value H (0x6a09e667UL) [32.le.32&]
00001896 1030 SHA256 [32.le.288&]
```

```
00002060 874 SHA256 Hash constant words K (0x428a2f98) [32.le.256]

- 3 signatures found in the file in 1 seconds
- done
```

We can see that Signsrch has successfully detected SHA-256, and the result will be recorded in Table 4 in our paper.

2.7 Where's Crypto

Where's Crypto is an IDA Pro-based tool.

2.7.1 Install the Tool

You can find the download link and instructions on how to install Where's Crypto in IDA Pro at `https://github.com/wheres-crypto/wheres-crypto`. You need MS Visual Studio, CMake, and Qt to run Where's Crypto.

2.7.2 Run the Tool

Firstly, load the binary program into IDA Pro, and then select Where's Crypto in plugin. You will see the detection analysis result. You can skip this tool if you do not have IDA Pro.

Tool		Cryptographic Algorithm									Micro-Benchmark				
		AES	DES	MD5	RC4	RC5	RSA	SHA1	SHA256	TEA	File	I/O	Math	Matrix	Network
DRACA	False Positive	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	False Negative	100%	100%	69%	0%	0%	0%	0%	0%	28%	0%	0%	0%	0%	0%
	True Positive	0%	0%	31%	0%	100%	0%	100%	0%	72%	0%	0%	0%	0%	0%
	True Negative	0%	0%	0%	100%	0%	100%	0%	100%	0%	100%	100%	100%	100%	100%
CryptoKnight	False Positive	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	74%	0%
	False Negative	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	True Positive	100%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%
	True Negative	0%	100%	0%	0%	100%	0%	100%	100%	100%	100%	100%	100%	26%	100%
Findcrypt2	False Positive	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	False Negative	100%	0%	69%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	True Positive	0%	100%	31%	0%	100%	0%	100%	100%	0%	0%	0%	0%	0%	0%
	True Negative	0%	0%	0%	100%	0%	100%	0%	0%	100%	100%	100%	100%	100%	100%
Signsrch	False Positive	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%
	False Negative	0%	100%	69%	0%	0%	0%	31%	0%	0%	0%	0%	0%	0%	0%
	True Positive	100%	0%	31%	0%	100%	0%	69%	100%	100%	0%	0%	0%	0%	0%
	True Negative	0%	0%	0%	100%	0%	100%	0%	0%	0%	100%	100%	100%	0%	100%
Where's Crypto	False Positive	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	False Negative	0%	0%	69%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	True Positive	100%	100%	31%	0%	0%	0%	100%	100%	100%	0%	0%	0%	0%	0%
	True Negative	0%	0%	0%	100%	100%	100%	0%	0%	0%	100%	100%	100%	100%	100%

Table 7. PERFORMANCE METRICS FOR TOOL EVALUATION: FALSE POSITIVES, FALSE NEGATIVES, TRUE POSITIVES, AND TRUE NEGATIVES

Tool	Aligot	Crypto-Hunt	Crypto-Knight	DRACA	FindCrypt2	FALKE-MC	HCD	Kerckhoff	PEiD KANAL	SignSrch	Softmax Classifier	Where's Crypto
ADLER32							*Not Available*		*Not Available*		✓	
AES (Rijndael)	✓	✓	✓	✓	✓	✓		✓		✓		✓
BASE64										✓	✓	
Blowfish			✓	✓	✓					✓	✓	
Camellia					✓							
CAST					✓					✓		
CAST-256				✓	✓					✓		
CRC32				✓	✓					✓	✓	
DES				✓	✓			✓		✓	✓	✓
EC										✓		
GOST					✓					✓		
HAVAL					✓					✓	✓	
MARS				✓	✓					✓		
MD2					✓					✓		
MD4					✓					✓		
MD5	✓	✓	✓	✓	✓			✓		✓	✓	✓
RC2				✓	✓					✓		
RC4	✓	✓	✓			✓		✓				
RC5				✓	✓					✓	✓	
RC6				✓	✓					✓	✓	
Ripemd-160				✓						✓		
RSA	✓	✓	✓					✓				
SAFER				✓	✓					✓		
SHA-1				✓	✓					✓	✓	✓
SHA-256					✓					✓	✓	✓
SHA-512					✓					✓		✓
SHARK					✓					✓		
Skipjack				✓	✓					✓		
Square					✓					✓		
TEA	✓	✓		✓						✓		
Tiger				✓	✓					✓		
Twofish				✓	✓					✓		
WAKE					✓					✓		
Whirlpool					✓					✓		
XTEA												✓

Table 8. CRYPTOGRAPHIC ALGORITHM DETECTION SUPPORT AS STATED IN THE PAPER OR DOCUMENTATION FOR EACH TOOL

Table 9. DETAILED EVALUATION RESULTS FOR EACH TOOL BASED ON OUR UNIFORM BENCHMARKING SUITE

Tool	Flag	Basic Cryptographic Function												Micro-Benchmark					Library											Large Project
		AES	DES	ECC	MD5	RC4	RC5	RSA	SHA1	SHA256	TEA	XTEA	XXTEA	File	I/O	Math	Matrix	Network	openssl	libgcrypt	libsodium	mbedTLS	gnuTLS	bzip2	zlib	ffmpeg	libgsm	libjpeg	libpng	signal
DRACA	-O0	○△□	○△□	○△□	●△■	○△□	●▲■	○△□	●▲■	○△□	●▲■	○△□	●△□	○△□	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲□
	-O1	○△□	○△□	○△□	○△■	○△□	●▲■	○△□	●▲■	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲□
	-O2	○△□	○△□	○△□	○△■	○△□	●▲■	○△□	●▲■	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲□
	-O3	○△□	○△□	○△□	○△■	○△□	●▲■	○△□	●▲■	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲□
	-Os	○△□	○△□	○△□	○△■	○△□	●▲■	○△□	●▲■	○△□	●▲■	●▲□	●▲□	○△□	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲□
	-Ofast	○△□	○△□	○△□	○△■	○△□	●▲■	○△□	●▲■	○△□	●▲■	●▲□	●▲□	○△□	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲□
	obs_sub	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○▲□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○▲□
	obs_fla	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○▲□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○▲□
	obs_bcf	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○▲□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○▲□
	obs_sub, obs_fla, obs_bcf	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○▲□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○▲□
	obs_sub, obs_fla, obs_bcf, -O3	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○▲□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○▲□
CryptoKnight	-O0	●▲□	○△□	○△□	○△□	○△□	○△□	●▲□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	●▲□	○△□	●▲□	○△□	○△□	○△□	○△□	○△□	○△□	●▲□	○△□	○△□	○△□	○△□
	-O1	●▲□	○△□	○△□	○△□	○△□	○△□	●▲□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	●▲□	○△□	●▲□	○△□	○△□	○△□	○△□	○△□	○△□	●▲□	○△□	○△□	○△□	○△□
	-O2	●▲□	○△□	○△□	○△□	○△□	○△□	●▲□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	●▲□	○△□	●▲□	○△□	○△□	○△□	○△□	○△□	○△□	●▲□	○△□	○△□	○△□	○△□
	-O3	●▲□	○△□	○△□	○△□	○△□	○△□	●▲□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	●▲□	○△□	●▲□	○△□	○△□	○△□	○△□	○△□	○△□	●▲□	○△□	○△□	○△□	○△□
	-Os	●▲□	○△□	○△□	○△□	○△□	○△□	●▲□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	●▲□	○△□	●▲□	○△□	○△□	○△□	○△□	○△□	○△□	●▲□	○△□	○△□	○△□	○△□
	-Ofast	●▲□	○△□	○△□	○△□	○△□	○△□	●▲□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	●▲□	○△□	●▲□	○△□	○△□	○△□	○△□	○△□	○△□	●▲□	○△□	○△□	○△□	○△□
	obs_sub	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○△□
	obs_fla	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○△□
	obs_bcf	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○△□
	obs_sub, obs_fla, obs_bcf	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○△□
	obs_sub, obs_fla, obs_bcf, -O3	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○△□
Findcrypt2	-O0	○△□	●▲■	○△□	●△■	○△□	●▲■	○△□	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	●▲■
	-O1	○△□	●▲■	○△□	○△■	○△□	●▲■	○△□	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	●▲■
	-O2	○△□	●▲■	○△□	○△■	○△□	●▲■	○△□	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	●▲■
	-O3	○△□	●▲■	○△□	○△■	○△□	●▲■	○△□	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	●▲■
	-Os	○△□	●▲■	○△□	○△■	○△□	●▲■	○△□	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	●▲■
	-Ofast	○△□	●▲■	○△□	○△■	○△□	●▲■	○△□	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	●▲■
	obs_sub	○△□	○▲□	○△□	○△□	○△□	○▲□	○△□	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○▲□
	obs_fla	○△□	○▲□	○△□	○△□	○△□	○▲□	○△□	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○▲□
	obs_bcf	○△□	○▲□	○△□	○△□	○△□	○▲□	○△□	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○▲□
	obs_sub, obs_fla, obs_bcf	○△□	○▲□	○△□	○△□	○△□	○▲□	○△□	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○▲□
	obs_sub, obs_fla, obs_bcf, -O3	○△□	○▲□	○△□	○△□	○△□	○▲□	○△□	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○▲□
HCD	-O0	○△■	○△□	○△□	○△■	○△□	○△□	○△□	○△■	○△■	○△■	○△■	○△■	○△□	○△□	○△□	○△□	○△□	○△■	○△■	○△■	○△□	○△□	○△■	○△■	○△■	○△□	○△□	○△□	○△■
	-O1	○△■	○△□	○△□	○△■	○△□	○△□	○△□	○△■	○△■	○△■	○△■	○△■	○△□	○△□	○△□	○△□	○△□	○△■	○△■	○△■	○△□	○△□	○△■	○△■	○△■	○△□	○△□	○△□	○△■
	-O2	○△■	○△□	○△□	○△■	○△□	○△□	○△□	○△■	○△■	○△■	○△■	○△■	○△□	○△□	○△□	○△□	○△□	○△■	○△■	○△■	○△□	○△□	○△■	○△■	○△■	○△□	○△□	○△□	○△■
	-O3	○△■	○△□	○△□	○△■	○△□	○△□	○△□	○△■	○△■	○△■	○△■	○△■	○△□	○△□	○△□	○△□	○△□	○△■	○△■	○△■	○△□	○△□	○△■	○△■	○△■	○△□	○△□	○△□	○△■
	-Os	○△■	○△□	○△□	○△■	○△□	○△□	○△□	○△■	○△■	○△■	○△■	○△■	○△□	○△□	○△□	○△□	○△□	○△■	○△■	○△■	○△□	○△□	○△■	○△■	○△■	○△□	○△□	○△□	○△■
	-Ofast	○△■	○△□	○△□	○△■	○△□	○△□	○△□	○△■	○△■	○△■	○△■	○△■	○△□	○△□	○△□	○△□	○△□	○△■	○△■	○△■	○△□	○△□	○△■	○△■	○△■	○△□	○△□	○△□	○△■
	obs_sub	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□
	obs_fla	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□
	obs_bcf	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□
	obs_sub, obs_fla, obs_bcf	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□
	obs_sub, obs_fla, obs_bcf, -O3	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□	○△□
PEiD KANAL	-O0	●▲■	●▲■	○△□	●△■	○△□	●▲■	○△□	●▲■	●▲■	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	●▲■	●▲■	○△□	○△□	○△□	●▲■
	-O1	●▲■	●▲■	○△□	○△■	○△□	●▲■	○△□	●▲■	●▲■	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	●▲■	●▲■	○△□	○△□	○△□	●▲■
	-O2	●▲■	●▲■	○△□	○△■	○△□	●▲■	○△□	●▲■	●▲■	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	●▲■	●▲■	○△□	○△□	○△□	●▲■
	-O3	●▲■	●▲■	○△□	○△■	○△□	●▲■	○△□	●▲■	●▲■	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	●▲■	●▲■	○△□	○△□	○△□	●▲■
	-Os	●▲■	●▲■	○△□	○△■	○△□	●▲■	○△□	●▲■	●▲■	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	●▲■	●▲■	○△□	○△□	○△□	●▲■
	-Ofast	●▲■	●▲■	○△□	○△■	○△□	●▲■	○△□	●▲■	●▲■	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	●▲■	●▲■	○△□	○△□	○△□	●▲■
	obs_sub	○▲□	○▲□	○△□	○△□	○△□	○▲□	○△□	○▲□	○▲□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○▲□	○▲□	○△□	○△□	○△□	○▲□
	obs_fla	○▲□	○▲□	○△□	○△□	○△□	○▲□	○△□	○▲□	○▲□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○▲□	○▲□	○△□	○△□	○△□	○▲□
	obs_bcf	○▲□	○▲□	○△□	○△□	○△□	○▲□	○△□	○▲□	○▲□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○▲□	○▲□	○△□	○△□	○△□	○▲□
	obs_sub, obs_fla, obs_bcf	○▲□	○▲□	○△□	○△□	○△□	○▲□	○△□	○▲□	○▲□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○▲□	○▲□	○△□	○△□	○△□	○▲□
	obs_sub, obs_fla, obs_bcf, -O3	○▲□	○▲□	○△□	○△□	○△□	○▲□	○△□	○▲□	○▲□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○▲□	○▲□	○△□	○△□	○△□	○▲□
Signsrch	-O0	●▲■	○△□	○△□	●△■	○△□	●▲■	○△□	●▲■	●▲■	●▲■	●▲■	●▲■	○△□	○△□	○△□	●▲■	○△□	○△□	●▲■	●▲■	○△□	○△□	●▲□	●▲■	●▲■	○△□	○△□	○△□	●▲■
	-O1	●▲■	○△□	○△□	○△■	○△□	●▲■	○△□	●▲■	●▲■	●▲■	●▲■	●▲■	○△□	○△□	○△□	●▲■	○△□	○△□	●▲■	●▲■	○△□	○△□	●▲□	●▲■	●▲■	○△□	○△□	○△□	●▲■
	-O2	●▲■	○△□	○△□	○△■	○△□	●▲■	○△□	●△■	●▲■	●▲■	●▲■	●▲■	○△□	○△□	○△□	●▲■	○△□	○△□	●▲■	●▲■	○△□	○△□	●▲□	●▲■	●▲■	○△□	○△□	○△□	●▲■
	-O3	●▲■	○△□	○△□	○△■	○△□	●▲■	○△□	○△■	●▲■	●▲■	●▲■	●▲■	○△□	○△□	○△□	●▲■	○△□	○△□	●▲■	●▲■	○△□	○△□	●▲□	●▲■	●▲■	○△□	○△□	○△□	●▲■
	-Os	●▲■	○△□	○△□	○△■	○△□	●▲■	○△□	●△■	●▲■	●▲■	●▲■	●▲■	○△□	○△□	○△□	●▲■	○△□	○△□	●▲■	●▲■	○△□	○△□	●▲□	●▲■	●▲■	○△□	○△□	○△□	●▲■
	-Ofast	●▲■	○△□	○△□	○△■	○△□	●▲■	○△□	○△■	●▲■	●▲■	●▲■	●▲■	○△□	○△□	○△□	●▲■	○△□	○△□	●▲■	●▲■	○△□	○△□	●▲□	●▲■	●▲■	○△□	○△□	○△□	●▲■
	obs_sub	○▲□	○△□	○△□	○△□	○△□	○▲□	○△□	○▲□	○▲□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○▲□	○△□	○△□	○▲□	○▲□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○▲□
	obs_fla	○▲□	○△□	○△□	○△□	○△□	○▲□	○△□	○▲□	○▲□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○▲□	○△□	○△□	○▲□	○▲□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○▲□
	obs_bcf	○▲□	○△□	○△□	○△□	○△□	○▲□	○△□	○▲□	○▲□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○▲□	○△□	○△□	○▲□	○▲□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○▲□
	obs_sub, obs_fla, obs_bcf	○▲□	○△□	○△□	○△□	○△□	○▲□	○△□	○▲□	○▲□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○▲□	○△□	○△□	○▲□	○▲□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○▲□
	obs_sub, obs_fla, obs_bcf, -O3	○▲□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○▲□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○▲□	○△□	○△□	○▲□	○▲□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○▲□
Where's Crypto	-O0	●▲■	●▲■	○△□	●△■	○△□	○△□	○△□	●▲■	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲■
	-O1	●▲■	●▲■	○△□	○△■	○△□	○△□	○△□	●▲■	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲■
	-O2	●▲■	●▲■	○△□	○△■	○△□	○△□	○△□	●▲■	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲■
	-O3	●▲■	●▲■	○△□	○△■	○△□	○△□	○△□	●▲■	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲■
	-Os	●▲■	●▲■	○△□	○△■	○△□	○△□	○△□	●▲■	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲■
	-Ofast	●▲■	●▲■	○△□	○△■	○△□	○△□	○△□	●▲■	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	●▲■	●▲■	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲■
	obs_sub	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○▲□
	obs_fla	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○▲□
	obs_bcf	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○▲□
	obs_sub, obs_fla, obs_bcf	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○▲□
	obs_sub, obs_fla, obs_bcf, -O3	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○▲□	○▲□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○▲□

Note: - ● represents the results of binaries compiled by GCC; ▲ represents the results of binaries compiled by LLVM; ■ represents the results of binaries compiled by MSVC.
- A full black shape indicates that the cryptographic function is detected in the binary, while an empty white shape indicates the opposite.
- Only LLVM supports obfuscation; flags with `obs_sub`, `obs_fla`, and `obs_bcf` are not available for GCC or MSVC (represented by dashed shapes).