MBA-Blast: Unveiling and Simplifying Mixed Boolean-Arithmetic Obfuscation

Binbin Liu, Junfu Shen, Jiang Ming, Qilong Zheng, Jing Li, Dongpeng Xu

Let us look at an MBA obfuscation example!

```
int fun(int x,int y,int z)
{
  int c;
  c = x+y;




  return c;
}
```

(a) Original program.

```
int fun(int x,int y,int z)
{
  int c;
  c = 4*(~x&y)-(x^y)-(x|y)
      +4*~(x|y)-~(x^y)-~y-
      (x|~y)+1+6*x+5*~z+
      (~(x^z))-(x|z)-2*~x-
      4*(~(x|z))-4*(x&~z)
      +3*(~(x|~z));

   return c;
}
```

(b) MBA obfuscated program.

Background

MBA Expression

Mixed Boolean-Arithmetic (MBA) expression mix Boolean operators (|, &, …) and arithmetic operations (+, -, *, …).

Definition 1. An MBA expression is :

$$\sum_{i \in I} a_i \boldsymbol{e_i}(\boldsymbol{x_1}, \ldots, \boldsymbol{x_t}),$$

a_i is a constant coefficient,

$\boldsymbol{e_i}(\boldsymbol{x_1}, \ldots, \boldsymbol{x_2})$ are bitwise expressions of variables $x_1, \ldots, x_t$.

$a_i \boldsymbol{e_i}(\boldsymbol{x_1}, \ldots, \boldsymbol{x_t})$ is called a term in the MBA expression.

2 ∗ (x | y) – (¬x & y) – (~x & y) – 9

X + y – (x&y) – 3*(x ^ y) + 5

MBA Obfuscation

Academia:

- Tigress: C source code obfuscator.

Industry:

- Cloakware
- Irdeto
- Quarkslab

Malware:

- VMProtect

MBA Deobfuscation

Existing software：

- Symbolic software： SagaMath, Wolfram Mathematica, Maple.
- SMT solver： Boolector, STP, Z3.

Related work：

- Arybo： normalizes MBA expressions to bit-level symbolic expressions.
- SSPAM： simplifies MBA expression by a pattern matching algorithm.
- Syntia： use program synthesis techniques to simplify MBA expressions.

MBA Deobfuscation

Existing software:

- Symbolic software： SagaMath, Wolfram Mathematica, Maple.
- SMT solver： Boolector, STP, Z3.

Related work:

- Arybo： normalizes MBA expressions to bit-level symbolic expressions.
- SSPAM： simplifies MBA expression by a pattern matching algorithm.
- Syntia： use program synthesis techniques to simplify MBA expressions.

MBA Deobfuscation

Existing software：

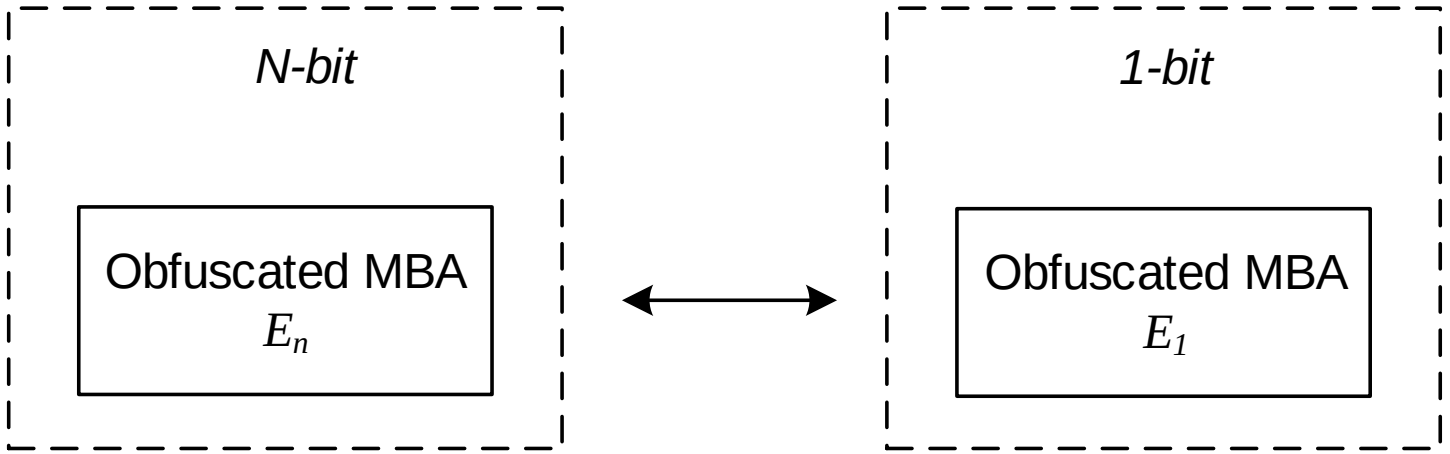
- Symbolic software： SagaMath, Wolfram Mathematica, Maple.
- SMT solver： Boolector, STP, Z3.

Related work：

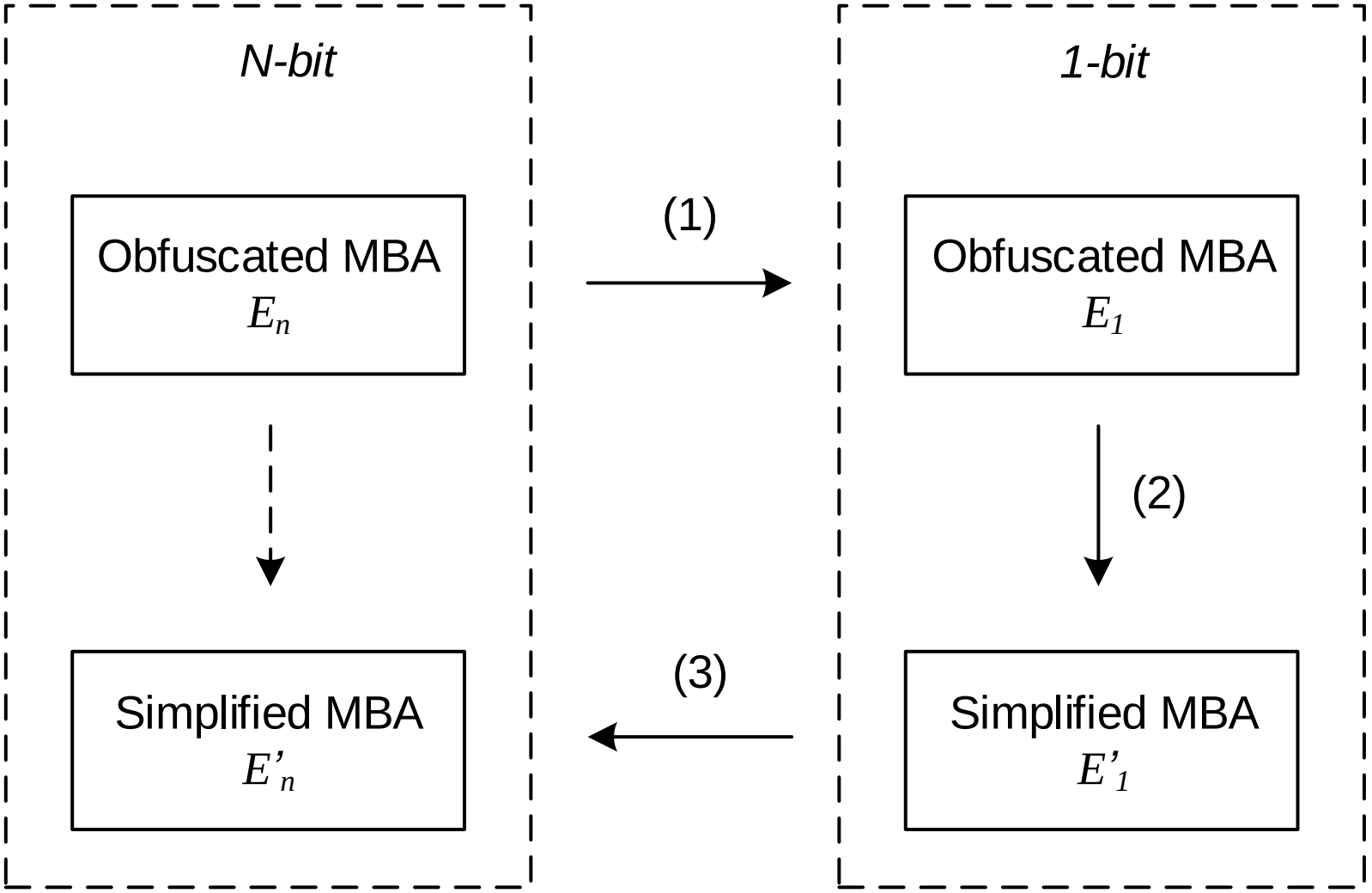
- Arybo： normalizes MBA expressions to bit-level symbolic expressions.
- SSPAM： simplifies MBA expression by a pattern matching algorithm.
- Syntia： use program synthesis techniques to simplify MBA expressions.

Our Approach

Two-way Feature



Logic Flow of MBA-Blast Simplification



2-Variable Truth Table

Truth Table	Boolean Expression
$[0,0,0,0]^T$	0
$[0,0,0,1]^T$	$x\& y$
$[0,0,1,0]^T$	$x\& \sim y$
$[0,0,1,1]^T$	x
$[0,1,0,0]^T$	$\sim x\& y$
… …	… …
$[1,0,1,1]^T$	$x\mid \sim y$
$[1,1,0,0]^T$	$\sim x$
$[1,1,0,1]^T$	$\sim x\mid y$
$[1,1,1,0]^T$	$\sim(x\& y)$
$[1,1,1,1]^T$	-1

2-Variable Boolean Normalization

Truth Table	Boolean Expression	MBA Expression
$[0,0,0,0]^T$	0	0
$[0,0,0,1]^T$	$x\& y$	$x\& y$
$[0,0,1,0]^T$	$x\& \sim y$	$x - (x\& y)$
$[0,0,1,1]^T$	x	x
$[0,1,0,0]^T$	$\sim x\& y$	$y - (x\& y)$
……	……	……
$[1,0,1,1]^T$	$x\| \sim y$	$-y + (x\& y) - 1$
$[1,1,0,0]^T$	$\sim x$	$-x - 1$
$[1,1,0,1]^T$	$\sim x\| y$	$-x + (x\& y) - 1$
$[1,1,1,0]^T$	$\sim(x\& y)$	$-(x\& y) - 1$
$[1,1,1,1]^T$	-1	-1

$\boldsymbol{e_i(x,y)}$
$= c_1 x + c_2 y + c_3(x\&y) - c_4$

MBA Simplification

$$\boldsymbol{e_i(x,y)} = c_1x + c_2y + c_3(x\&y) - c_4$$

$$\sum_{i\in I} a_i \boldsymbol{e_i(x,y)} = \sum_{i\in I} a_i * (c_{1i}x + c_{2i}y + c_{3i}(x\&y) - c_{4i}) = \boldsymbol{C_1x + C_2y + C_3(x\&y) - C_4}$$

MBA Simplification Algorithm

MBA expression

$$2*(x|y) - (\sim x\&y) - (x\&\sim y) - 2*(x\&y)$$

MBA Simplification Algorithm

MBA expression

Boolean Normalization

$$\boldsymbol{e_i(x,y)} = c_1x + c_2y + c_3(x\&y) - c_4$$

2*(x|y) – (~x&y) – (x&~y) – 2*(x&y)

2*(x + y – (x&y)) – (y – (x&y)) – (x – (x&y)) – 2*(x&y)

MBA Simplification Algorithm

MBA expression

Boolean Normalization

$$\boldsymbol{e_i}(\boldsymbol{x},\boldsymbol{y}) = c_1x + c_2y + c_3(x\&y) - c_4$$

Arithmetic Reduction

$$\sum_{i\in I} a_i \boldsymbol{e_i}(\boldsymbol{x},\boldsymbol{y})$$
$$= \sum_{i\in I} a_i * (c_{1i}x + c_{2i}y + c_{3i}(x\&y) - c_{4i})$$
$$= C_1x + C_2y + C_3(x\&y) - C_4$$

2*(x|y) – (~x&y) – (x&~y) – 2*(x&y)

2*(x + y – (x&y)) – (y – (x&y)) – (x – (x&y)) – 2*(x&y)

2*x + 2*y – 2*(x&y) – y + (x&y) – x + (x&y) – 2*(x&y)
= x + y – 2*(x&y)

MBA Simplification Algorithm

MBA expression

Boolean Normalization

$$\boldsymbol{e_i}(\boldsymbol{x},\boldsymbol{y}) = c_1x + c_2y + c_3(x\&y) - c_4$$

Arithmetic Reduction

$$\sum_{i\in I} a_i \boldsymbol{e_i}(\boldsymbol{x},\boldsymbol{y})$$
$$= \sum_{i\in I} a_i * (c_{1i}x + c_{2i}y + c_{3i}(x\&y) - c_{4i})$$
$$= C_1x + C_2y + C_3(x\&y) - C_4$$

Optimization

Simple bitwise expression?

2*(x|y) – (~x&y) – (x&~y) – 2*(x&y)

2*(x + y – (x&y)) – (y – (x&y)) – (x – (x&y)) – 2*(x&y)

2*x + 2*y – 2*(x&y) – y + (x&y) – x + (x&y) – 2*(x&y)
= x + y – 2*(x&y)

x^y

Evaluation

MBA Dataset

Setup

- 10,000 MBA expressions
  - Length of variables
  - Number of variables
  - Number of terms
- Intel Xeon CPU and 64GB RAM
- Compared with peer tools
  - Arybo
  - SSPAM
  - Syntia
- Z3 solver for checking simplified result
  - Timeout threshold: five hours

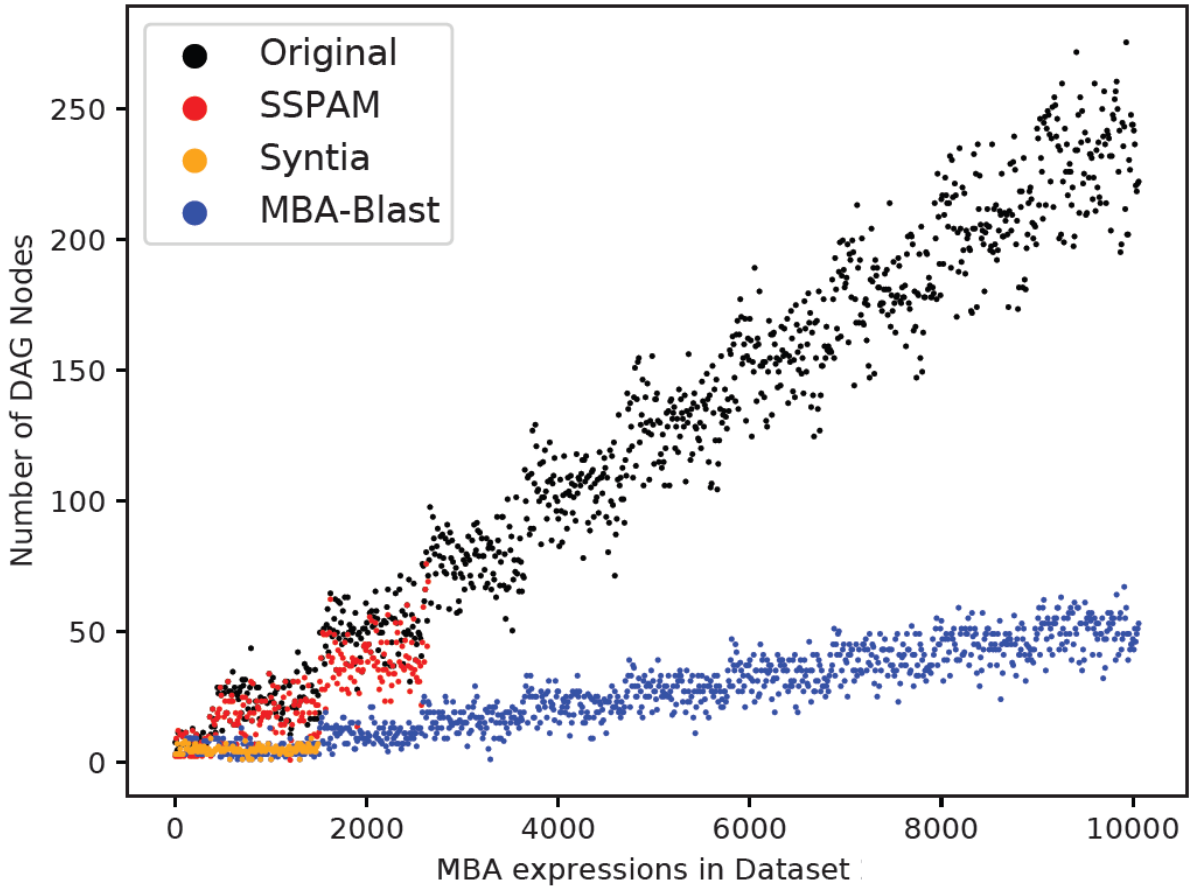
MBA Dataset

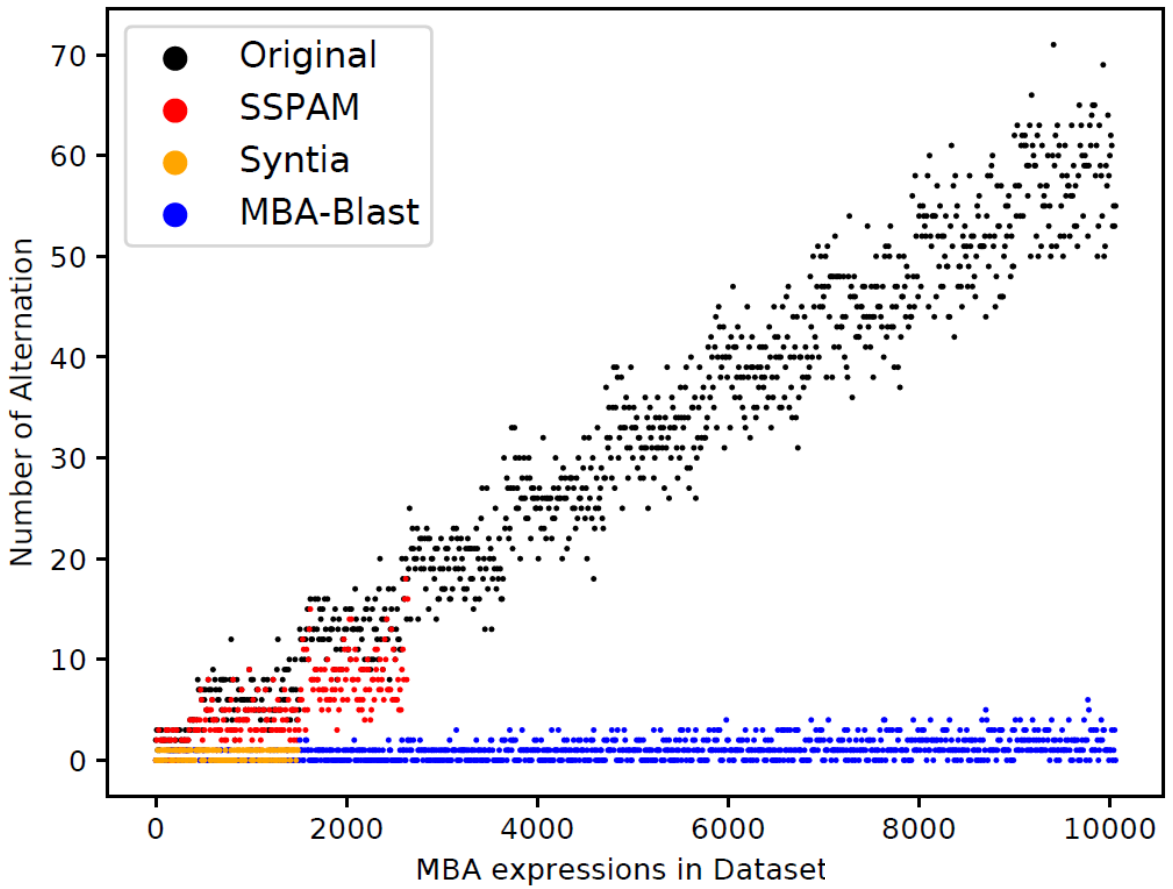
Simplification Results

Method	Simplification Results				Average Processing Time (Seconds/Sample)
	Yes	No	T.O	Ratio(%)	
Arybo	431	0	9569	4.3	640.7
SSPAM	2550	0	7450	25.5	438.2
Syntia	1438	8562	0	14.1	9.3
MBA-Blast	**10,000**	**0**	**0**	**100.0**	**0.05**

MBA Dataset

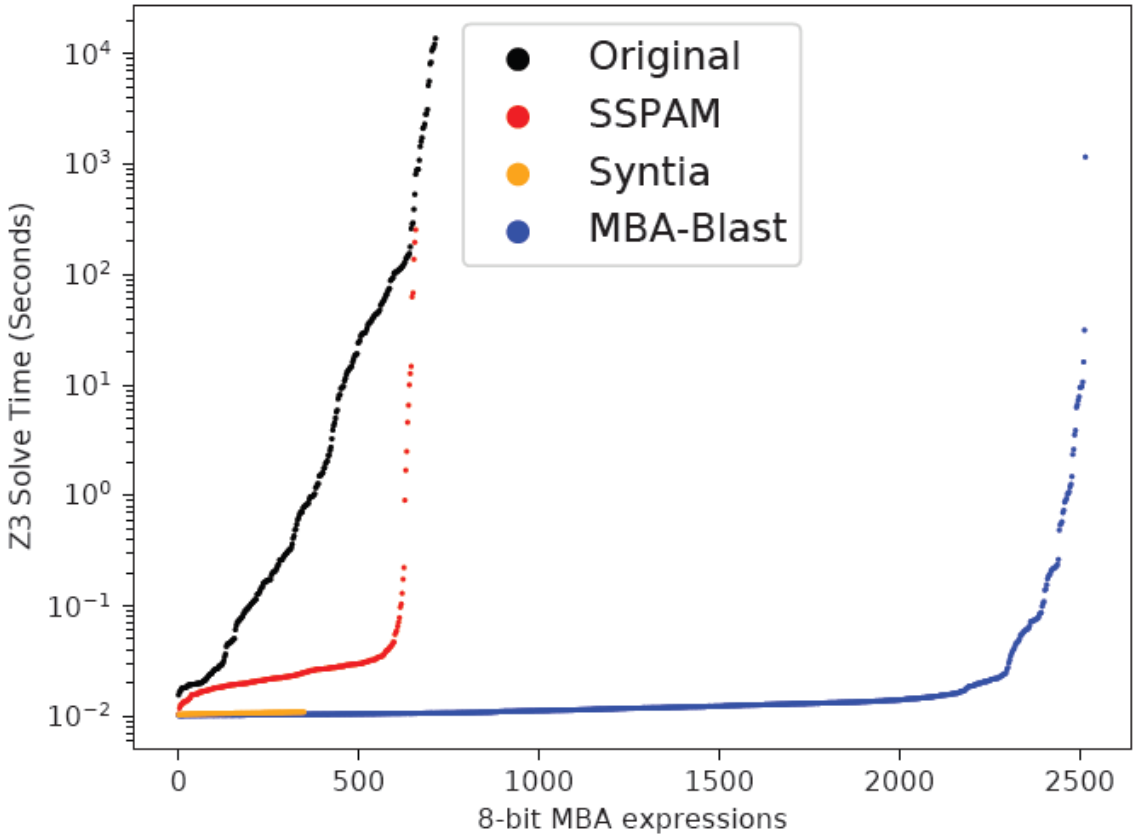
Complexity Metrics

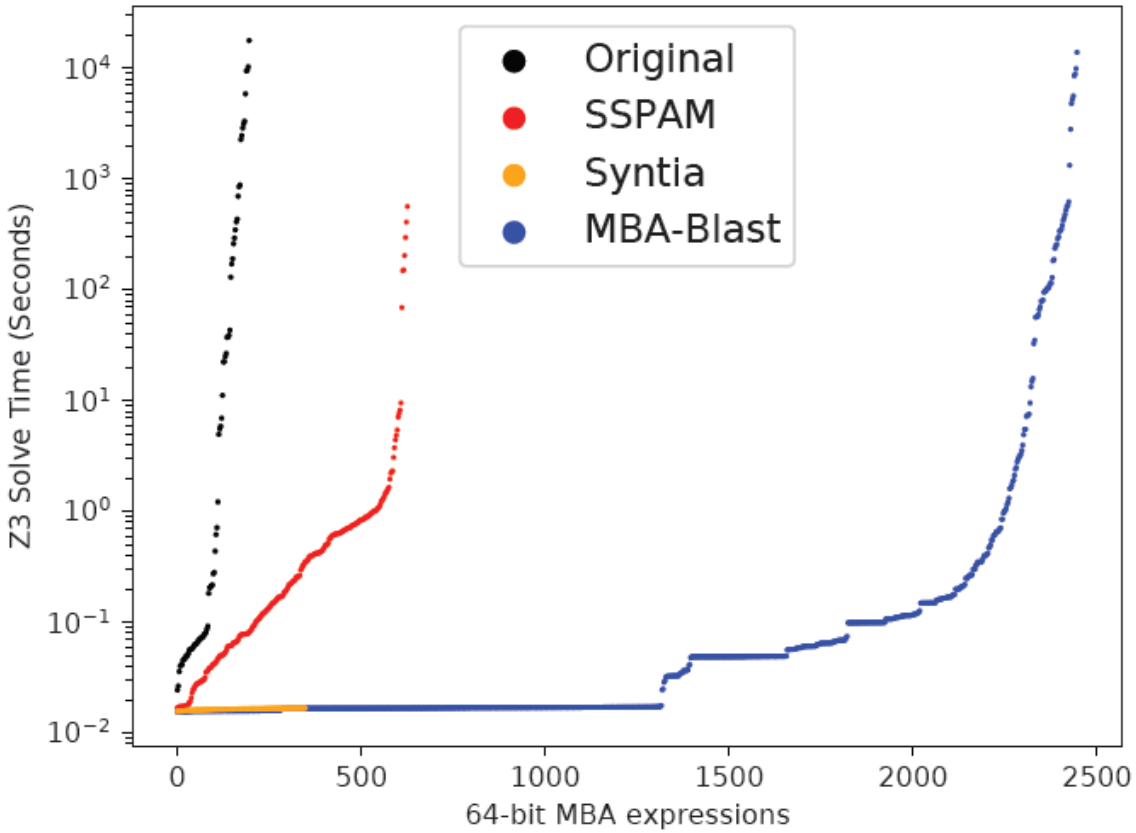




MBA Dataset

Z3 Solving Time





Malware Dataset

Vmprotect: 132 malware samples from VirusTotal

Category	Number	# with MBA	MBA Expressions
Trojan	36	30	41
Virus	33	26	40
Malware	33	26	41
Riskware	8	7	11
CoinMiner	7	6	9
Backdoor	4	2	3
ADware	4	3	4
Ransomware	3	3	5
Spyware	2	1	2
Others	2	1	1
Total	132	105	157

Malware Dataset

Example: $\mathrm{x} - y = \sim(\sim\mathrm{x} + \ \mathrm{y})\& \sim(\sim x + y)$

$\sim(\sim\mathrm{x} + \ \mathrm{y})\& \sim(\sim x + y)$

$= \sim t\& \sim t \qquad\qquad \sim x + y \to t$

$= -t - 1$

$= -\sim\mathrm{x} - \mathrm{y} - 1 \qquad\qquad t \to \sim x + y$

$= \mathrm{x} - y$

Case Study

Ransomware Sample

$$(\mathrm{K}|{\sim}\mathrm{C}) + ({\sim}K|C) - 2 * {\sim}(K|C) - 2 * (K\&C)$$

$$= -C + (K\&C) - 1 + (-K + (K\&C) - 1)$$
$$-2 * (-K - C + (K\&C) - 1) - 2 * (K\&C)$$
$$= K + C - 2 * (K\&C)$$
$$= (K\hat{}C)$$

Conclusion

MBA-Blast is
the most generic and efficient
MBA deobfuscation technique

Binbin.liu@unh.edu
Dongpeng.xu@unh.edu

https://github.com/softsec-unh/MBA-Blast