Evaluating Approaches for Identifying Cryptographic Functions in Binaries

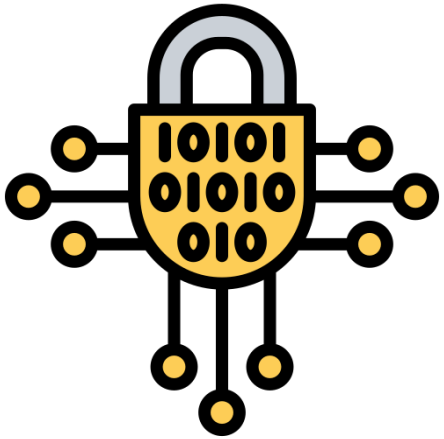
Yongming Fan[1], Priyam Biswas[2], and Christina Garman[1]

[1] Purdue University, [2] Intel

PURDUE UNIVERSITY | Department of Computer Science

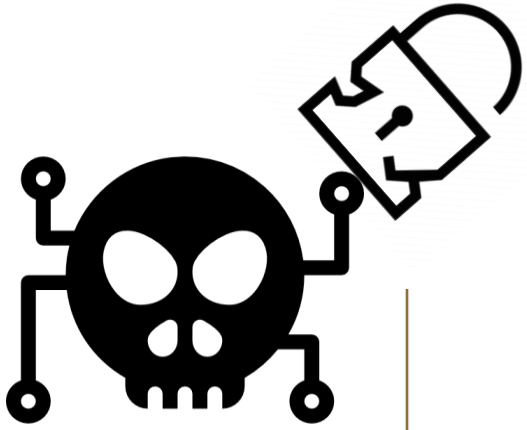
INTRODUCTION

- Cryptographic functions are instrumental in securing communications and systems



- They are mis-used, mis-implemented, or created in an ad-hoc manner
- It is also used for malicious purposes such as hiding payloads in malware
- Occur in closed-source code or binary applications



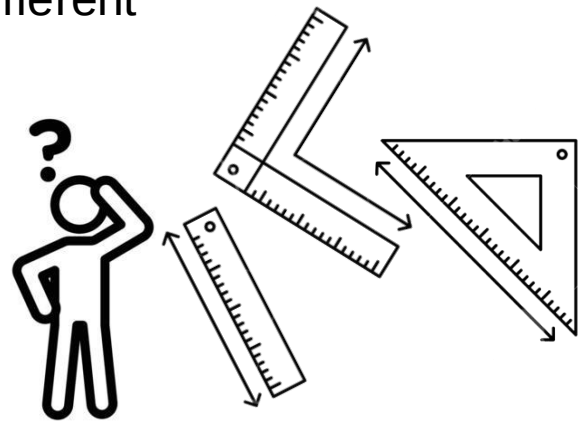


INTRODUCTION

- General purpose binary analysis and function identification techniques could help solve these problems

- A variety of work has focused on a subset of this space
  - Developing techniques and tools that are specifically tailored to identifying different cryptographic primitives in binary applications

- Lacks consistent means of evaluation or comparisons across tools



INTRODUCTION

- How can we design a better experiment to effectively evaluate cryptographic function detection tools
  - Understand current status, performance, and existing gaps
  - Gain better insights into this field



INTRODUCTION

- How can we design a better experiment to effectively evaluate cryptographic function detection tools
  - Understand current status, performance, and existing gaps
  - Gain better insights into this field



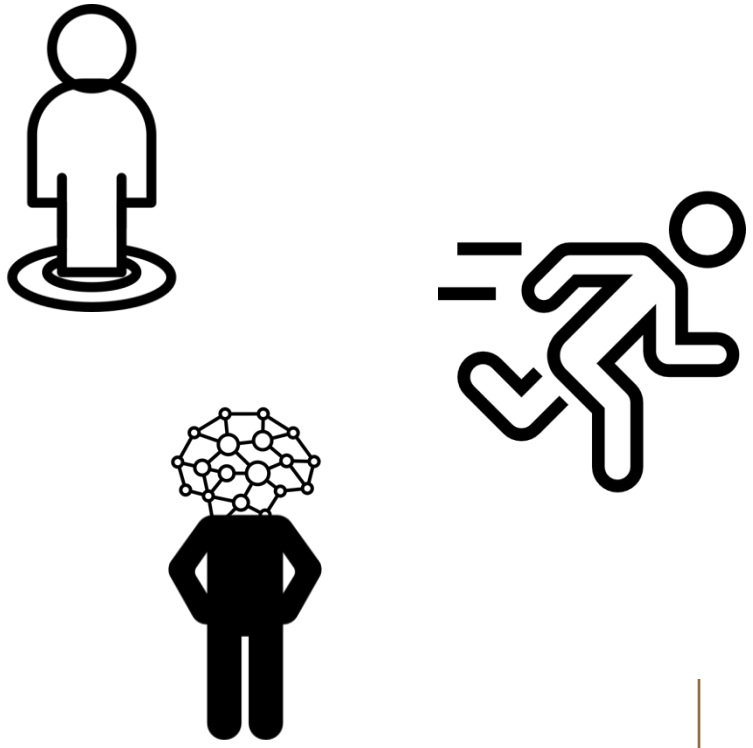
- Propose a novel analysis framework aimed at facilitating a comprehensive evaluation of cryptographic function detection tools' replicability

- Focus on comprehensive replication and reproduction studies
  - Existing work on identifying cryptography code in binary applications

BACKGROUND
CATEGORIZATION OF DETECTION APPROACHES

- Scalability and performance of an identification approach all rely on the underlying techniques used

- Categorization of detection approaches
  - Static Approaches
  - Dynamic Approaches
  - Machine Learning Based Approaches



BACKGROUND
CRYPTOGRAPHIC FEATURES

- Cryptographic functions have their own set of characteristics

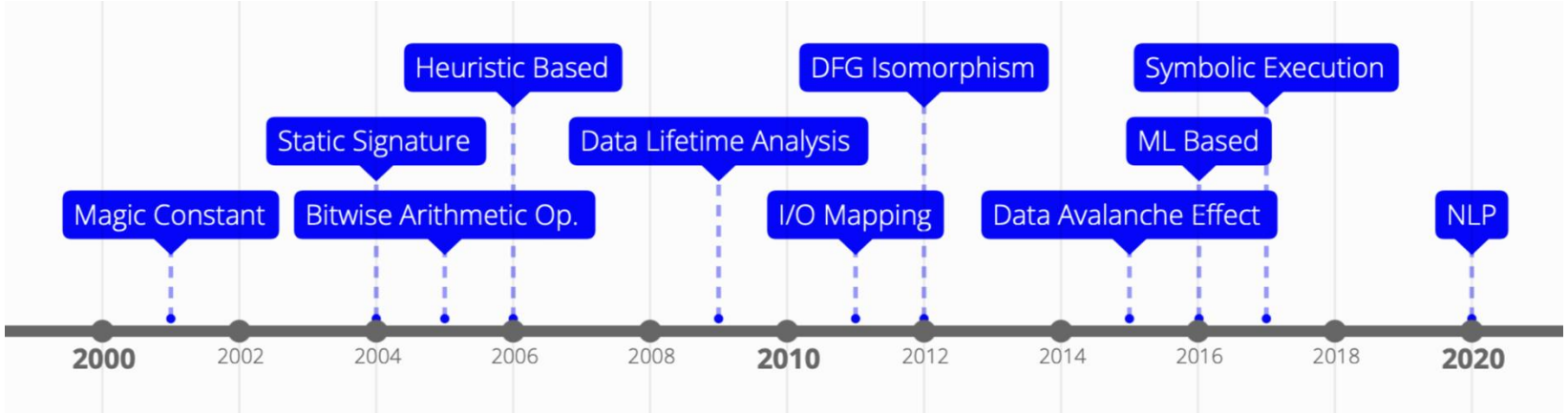
- Many of these features only pertain to certain algorithms and cannot be used to identify others
  - Magic Constants
  - Presence of Loops
  - Changes in Entropy
  - I/O Mapping
  - Data-Flow Isomorphism
  - S-Box
  - Instruction Sequences



BACKGROUND
CRYPTOGRAPHIC FEATURES

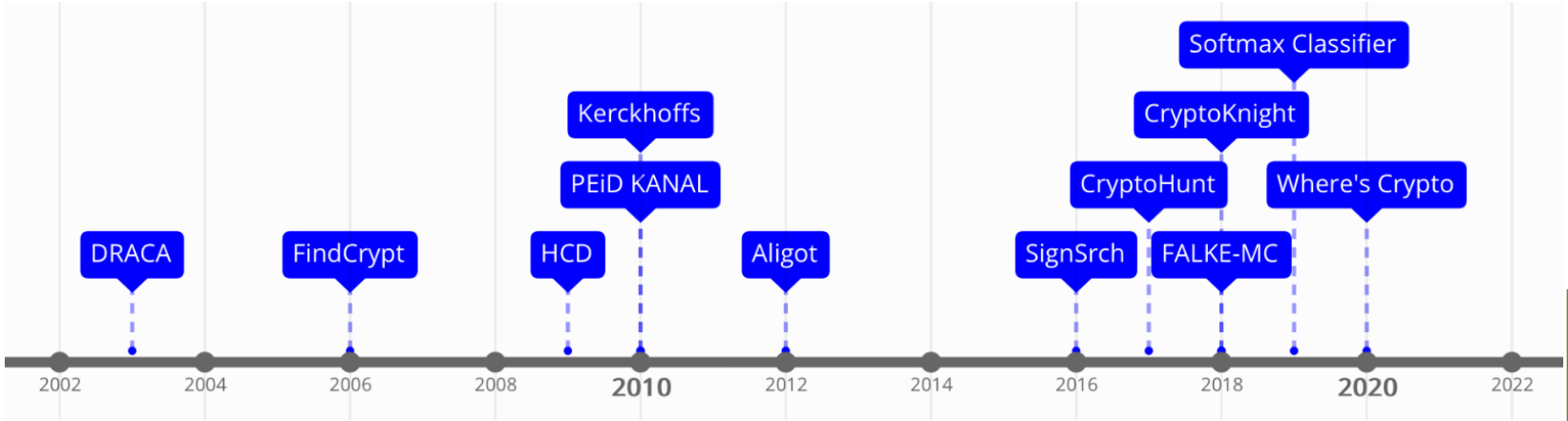
- Cryptographic functions have their own set of characteristics

- Many of these features only pertain to certain algorithms and cannot be used to identify others



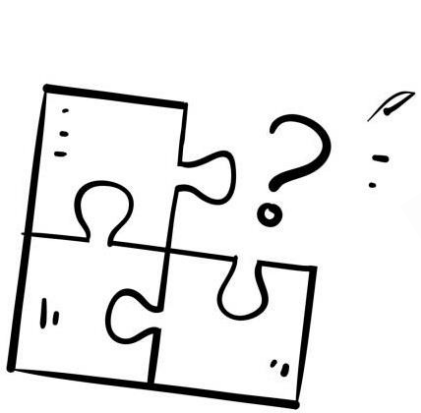
CRYPTOGRAPHIC ALGORITHM DETECTION TOOLS

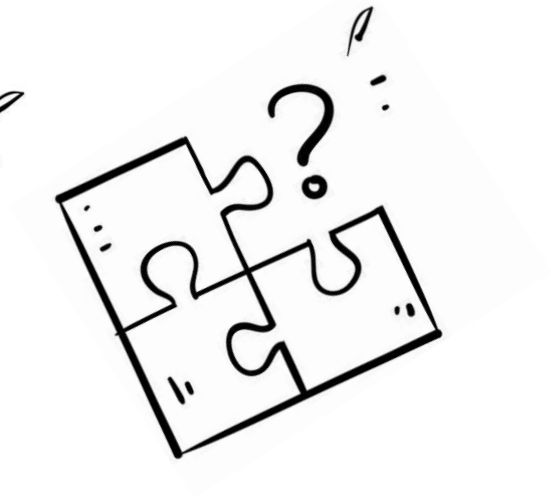
- We perform our reproduction and replication studies on all existing work that we were able to find from 2000 until now:
  - Aligot
  - CryptoHunt
  - CryptoKnight
  - DRACA
  - FindCrypt2
  - FALKE-MC
  - HCD
  - Kerckhoff
  - PEiD KANAL
  - SignSrch
  - Softmax Classifier
  - Where's Crypto



ANALYSIS FRAMEWORK FOR OPEN-SOURCE DETECTION TOOLS

- Previous tools have been evaluated only with certain basic cryptographic algorithms
- Not on the same framework/benchmark
- Lack evaluation for false positives, real world applications, and large scale projects







ANALYSIS FRAMEWORK FOR OPEN-SOURCE DETECTION TOOLS

PURDUE UNIVERSITY | Department of Computer Science

	$\mathcal{B}_1$	$\mathcal{B}_2$	*Storm**	*SBank**	$\mathcal{B}_3$	*Sal**	$\mathcal{B}_4$	$\mathcal{B}_5$†	$\mathcal{B}_6$	$\mathcal{B}_7$†	*Wal**	$\mathcal{B}_8$†
Aligot	TEA	TEA	RTEA	RTEA	RC4	RC4	AES	AES	MD5	MD5	AES, MD5	MOD MUL
CryptoSearcher	TEA	×	×	×	×	×	AES	×	MD5	×	×	×
Draca	TEA	×	×	TEA	×	×	×	×	MD5	SHA-1	×	×
Findcrypt2	×	×	×	×	×	×	AES	×	MD4	×	×	×
H&C Detector	TEA	×	×	TEA	×	×	×	×	MD5	SHA-1	×	×
Kerckhoffs	×	×	×	×	×	×	×	×	×	×	×	×
PEiD KANAL	TEA	×	×	TEA	×	×	AES	×	MD5	×	×	×
Signsrch	TEA	×	×	TEA	×	×	AES	×	×	×	×	×
SnDCryptoS	×	×	×	×	×	×	AES	×	MD5	×	×	×

	AES	RC4	BLF	MD5	RSA	R/A
AES	13	0	0	0	0	0
RC4	0	12	1	0	0	0
BLF	0	0	12	0	0	1
MD5	0	1	0	12	0	0
RSA	0	0	0	0	13	0
R/A	0	0	0	0	0	13

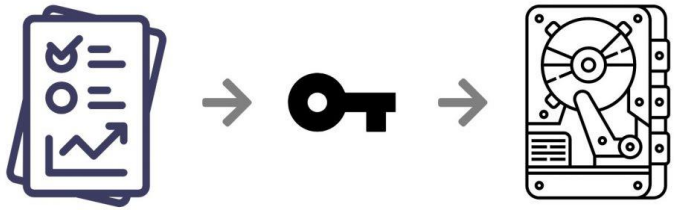
Algorithm signature	dropbear	libcrypto.so.1.1	libmbedcrypto.so.2.16.3[1]	libnettle.so.7.0[2]
size	145 KB	1,735 KB	197 KB	237 KB
analysis time	6m44s	39m47s	6m56s	11m32s
SHA1				
sha1	✓ Unlabeled[3]	✓ SHA1_Update	✓ sha1_update_ret	✓ sha1_compress
bl.perm.	✓ Unlabeled[3]	✓ SHA1_Update	✓ sha1_update_ret	✓ sha1_update[4]
SHA256				
bl.perm.	✓ Unlabeled[3]	✓ SHA256_Update[5]	✓ sha256_update_ret	✓ sha256_update[4,5]
AES				
aes	✓ Unlabeled[3]	✓ AES_encrypt	✓ aes_encrypt	✓ aes_encrypt_armv6
MD4				
bl.perm.	N/A	✓ MD4_Update	N/A	✓ md4_update[4]
MD5				
md5	N/A	✓ MD5_Update	✓ md5_update_ret	✓ hmac_md5_update
bl.perm.	N/A	✓ MD5_Update	✓ md5_update_ret	✓ hmac_md5_update
RIPEMD160				
bl.perm.	N/A	✓ RIPEMD160_Update	N/A	✓ hmac_ripemd160_update
SHA512				
bl.perm.	N/A	✓ SHA512_Update[5]	✓ sha512_process[5]	✓ sha512_update[5]

Crypto Lib	Algo	Obf	Findcrypto2	Signsrch	CryptoSearcher	DFGIsom	Kerchkhoffs	Aligot	CryptoHunt
OpenSSL	TEA	N		✓	✓	✓	✓	✓	✓
		O1		✓	✓		✓	✓	✓
		O2							✓
	AES	N	✓	✓	✓	✓		✓	✓
		O1	✓	✓	✓			✓	✓
		O2							✓
	RC4	N						✓	✓
		O1						✓	✓
		O2							✓
	MD5	N			✓	✓		✓	✓
		O1			✓			✓	✓
		O2							✓
	RSA	N							✓
		O1							✓
		O2							✓
Libgcrypt	TEA*	N		✓	✓	✓	✓	✓	✓
		O1		✓	✓		✓	✓	✓
		O2							✓
	AES	N	✓	✓		✓			✓
		O1	✓	✓					✓
		O2							✓
	RC4	N						✓	✓
		O1						✓	✓
		O2							✓
	MD5	N			✓	✓		✓	✓
		O1			✓			✓	✓
		O2							✓
	RSA	N							✓
		O1							✓
		O2							✓

EXPERIMENT DESIGN
BENCHMARKING FRAMEWORK

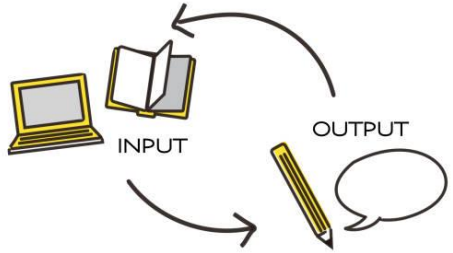
PURDUE UNIVERSITY | Department of Computer Science

- Basic cryptographic functions
  - AES256
  - DES
  - ECC
  - MD5
  - RC4
  - RC5
  - RSA
  - SHA1
  - SHA256
  - TEA
  - XTEA
  - XXTEA
- Cryptographic function or cryptographic algorithm refer to the entirety of a specific scheme or class of algorithms



EXPERIMENT DESIGN
BENCHMARKING FRAMEWORK

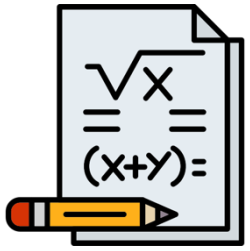
- Basic cryptographic functions



- Microbenchmarks
  - File Manipulation
  - Heavy Math
  - IO Program
  - Matrix
  - Network









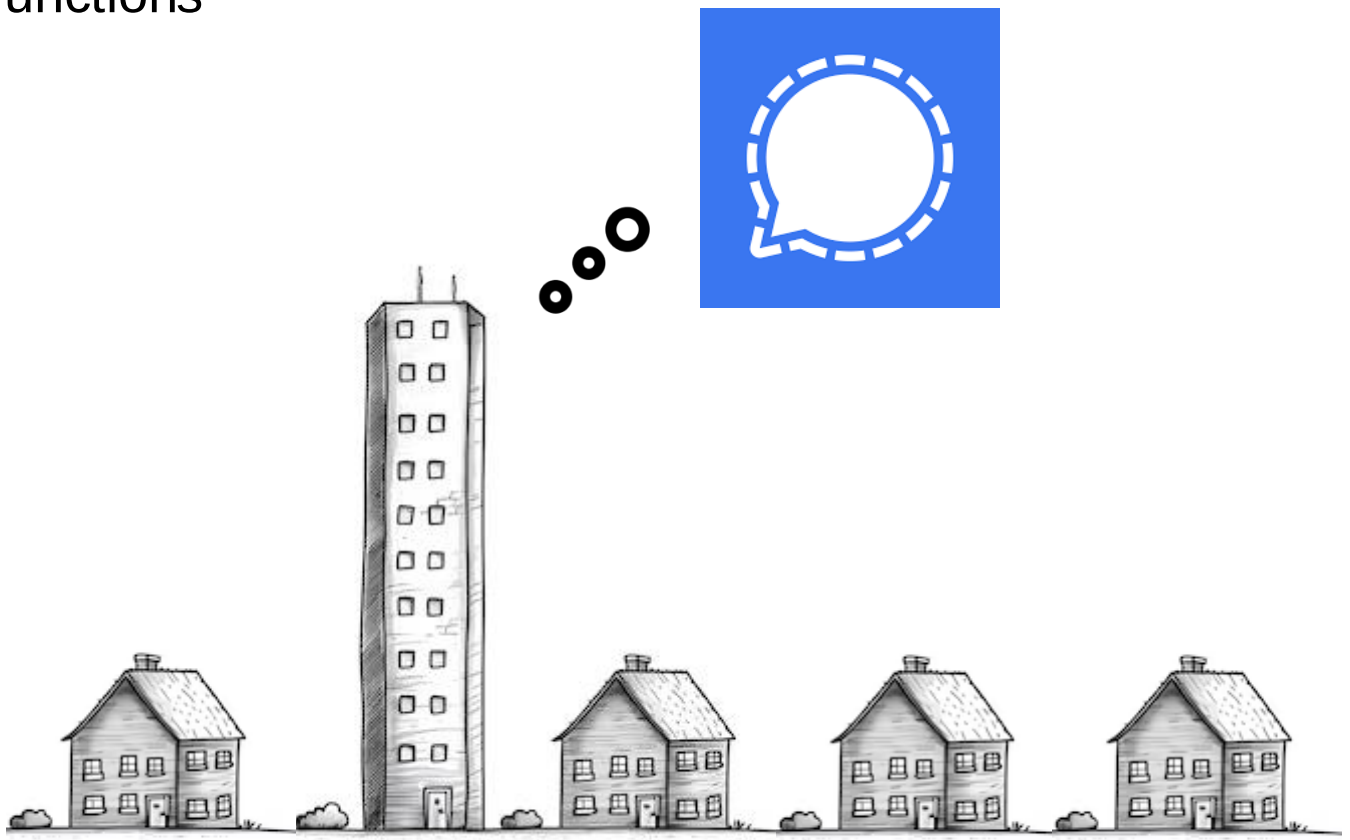
EXPERIMENT DESIGN
BENCHMARKING FRAMEWORK

- Basic cryptographic functions
- Microbenchmarks
- Libraries
  - openssl (3.3.1)
  - libgcrypt (1.8.11)
  - libsodium (1.0.20)
  - mbedTLS (3.6.1)
  - gnuTLS (3.8.6)
  - bzip2 (1.0.8)
  - zlib (1.3.1)
  - ffmpeg (7.0.2)
  - libgsm (1.0.17)
  - libjpeg (6b)
  - libpng (1.6.43)



EXPERIMENT DESIGN
BENCHMARKING FRAMEWORK

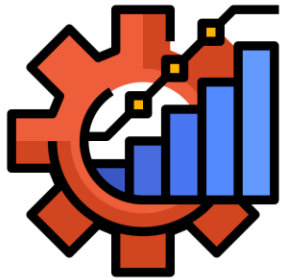
- Basic cryptographic functions
- Microbenchmarks
- Libraries
- Large project
  - Signal



EXPERIMENT DESIGN
EVALUATION METRICS

- Based on optimization levels
  - -O0, -O1, -O2, -O3, -Os, and -Ofast
- Based on different combinations of obfuscation mechanisms:
  - Control-flow obfuscation
  - Data-flow obfuscation
  - Modified versions of algorithms
  - Layout Obfuscation
  - Combination of all prior techniques
- Based on different compilers: Evaluate based on different compilers
  - GCC, CLANG, MSVC

Crypto Lib	Algo	Obf
OpenSSL	TEA	N O1 O2
	AES	N O1 O2
	RC4	N O1 O2
	MD5	N O1 O2
	RSA	N O1 O2





EVALUATION RESULT

PURDUE UNIVERSITY | Department of Computer Science

Tool	Flag	Basic Cryptographic Function											
		AES	DES	ECC	MD5	RC4	RC5	RSA	SHA1	SHA256	TEA	XTEA	XXTEA
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	○△⬚	○△⬚	○△⬚	○△⬚	○△⬚	○▲⬚	○△⬚	○▲⬚	○△⬚	○▲⬚	○▲⬚	○▲⬚

Micro-Benchmark					Library											Large Project
File	I/O	Math	Matrix	Network	openssl	libgcrypt	libsodium	mbedTLS	gnuTLS	bzip2	zlib	ffmpeg	libgsm	libjpeg	libpng	signal
○△□	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲□
○△□	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲□
○△□	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲□
○△□	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲□
○△□	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲□
○△□	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲□
○△⬚	○△⬚	○△⬚	○△⬚	○△⬚	○△⬚	○▲⬚	○△⬚	○△⬚	○△⬚	○△⬚	○△⬚	○▲⬚	○△⬚	○△⬚	○△⬚	○▲⬚
○△⬚	○△⬚	○△⬚	○△⬚	○△⬚	○△⬚	○▲⬚	○△⬚	○△⬚	○△⬚	○△⬚	○△⬚	○▲⬚	○△⬚	○△⬚	○△⬚	○▲⬚
○△⬚	○△⬚	○△⬚	○△⬚	○△⬚	○△⬚	○▲⬚	○△⬚	○△⬚	○△⬚	○△⬚	○△⬚	○▲⬚	○△⬚	○△⬚	○△⬚	○▲⬚
○△⬚	○△⬚	○△⬚	○△⬚	○△⬚	○△⬚	○▲⬚	○△⬚	○△⬚	○△⬚	○△⬚	○△⬚	○▲⬚	○△⬚	○△⬚	○△⬚	○▲⬚
○△⬚	○△⬚	○△⬚	○△⬚	○△⬚	○△⬚	○▲⬚	○△⬚	○△⬚	○△⬚	○△⬚	○△⬚	○▲⬚	○△⬚	○△⬚	○△⬚	○▲⬚

EVALUATION RESULT

PURDUE UNIVERSITY | Department of Computer Science

Tool	Flag	Basic Cryptographic Function											
		AES	DES	ECC	MD5	RC4	RC5	RSA	SHA1	SHA256	TEA	XTEA	XXTEA
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	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○▲□	○△□	○▲□	○▲□	○▲□

Micro-Benchmark					Library										Large Project
File	I/O	Math	Matrix	Network	openssl	libgcrypt	libsodium	mbedTLS	gnuTLS	bzip2	zlib	ffmpeg	libgsm	libjpeg	libpng	signal
○△□	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲□
○△□	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲□
○△□	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲□
○△□	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲□
○△□	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲□
○△□	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	○△□	○△□	●▲■	○△□	○△□	○△□	●▲□
○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○▲□
○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○▲□
○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○▲□
○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○▲□
○△□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○△□	○△□	○▲□	○△□	○△□	○△□	○▲□

META DISCUSSION

- Did you use any experimentation artifacts from the community?
- Did you attempt to replicate or reproduce earlier research?
- What insights can be gained from your methodology?
- What challenges did you face, and how did you overcome them?
- Did you produce any intermediate or unsuccessful results?
- Have you shared your experimental artifacts with the community?

REPRODUCTION AND REPLICATION

- Reproduction and Replication

- Reproduction
  - Use the same measurement procedure, system setting, and operating conditions as the tool's origin

- Replication
  - Rebuild each tool and run it across different system environments using our newly developed evaluation benchmark as the test cases



DISCUSSION

- Standardized Evaluation
- Ability vs. Performance
- User Experience & Reliability





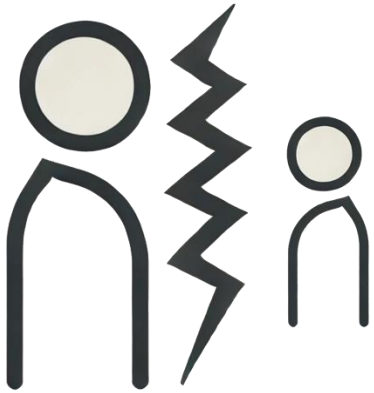


DIFFICULTY

- User-friendly cryptographic detection tools
  - Do not require developers to compile and build the tools
  - Exhibit lower performance and utility

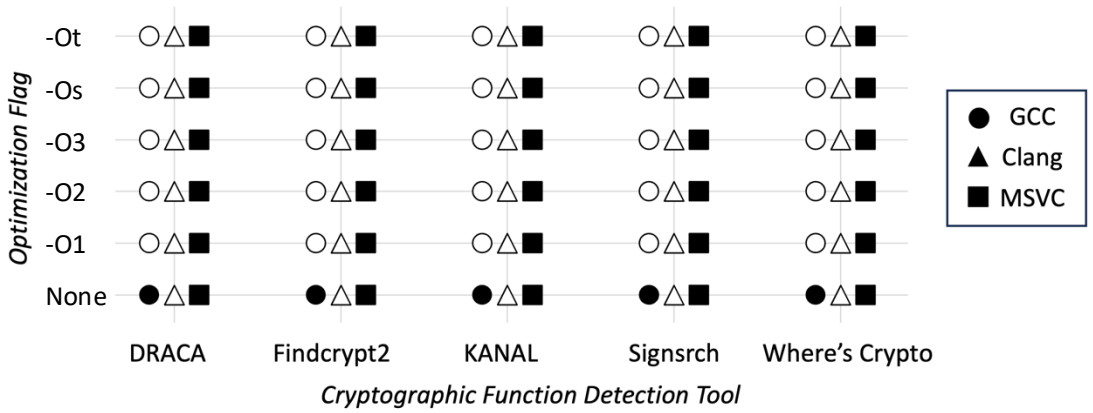


- Dependent libraries requirements
  - IDA Pro
  - Intel Pin



INTERMEDIATE OR UNSUCCESSFUL RESULTS

Tool		Cryptographic Algorithm								
		AES	DES	MD5	RC4	RC5	RSA	SHA1	SHA256	TEA
DRACA	False Positive	0%	0%	0%	0%	0%	0%	0%	0%	0%
	False Negative	100%	100%	69%	0%	0%	0%	0%	0%	28%
	True Positive	0%	0%	31%	0%	100%	0%	100%	0%	72%
	True Negative	0%	0%	0%	100%	0%	100%	0%	100%	0%
CryptoKnight	False Positive	0%	0%	0%	100%	0%	0%	0%	0%	0%
	False Negative	0%	0%	100%	0%	0%	0%	0%	0%	0%
	True Positive	100%	0%	0%	0%	0%	100%	0%	0%	0%
	True Negative	0%	100%	0%	0%	100%	0%	100%	100%	100%
Findcrypt2	False Positive	0%	0%	0%	0%	0%	0%	0%	0%	0%
	False Negative	100%	0%	69%	0%	0%	0%	0%	0%	0%
	True Positive	0%	100%	31%	0%	100%	0%	100%	100%	0%
	True Negative	0%	0%	0%	100%	0%	100%	0%	0%	100%
Signsrch	False Positive	0%	0%	0%	0%	0%	0%	0%	0%	0%
	False Negative	0%	100%	69%	0%	0%	0%	31%	0%	0%
	True Positive	100%	0%	31%	0%	100%	0%	69%	100%	100%
	True Negative	0%	0%	0%	100%	0%	100%	0%	0%	0%
Where's Crypto	False Positive	0%	0%	0%	0%	0%	0%	0%	0%	0%
	False Negative	0%	0%	69%	0%	0%	0%	0%	0%	0%
	True Positive	100%	100%	31%	0%	0%	0%	100%	100%	100%
	True Negative	0%	0%	0%	100%	100%	100%	0%	0%	0%



FUTURE WORKS

- Provide better information for current tool
  - Detailed performance evaluation from our R+R study
  - Updated technical instruction for each cryptographic function detection tool



- Finding gap in this field for future research
  - Reduce false positive and false negative
  - Reduce operating system limitation
  - Support modern cryptographic algorithm
  - Minimize impact of compiler
  - Minimize impact of optimization/obfuscation



THANK YOU!

Yongming Fan: fan322@purdue.edu
Priyam Biswas: priyam.biswas@intel.com
Christina Garman: clg@purdue.edu

https://github.com/BARC-Purdue/CryptoBinary

PURDUE UNIVERSITY | Department of Computer Science