

# PENGFEI GAO

(+86)18621312419  $\diamond$  gaopf@shanghaitech.edu.cn  $\diamond$  cumt-gpf.github.io

## EDUCATION

---

### Ph.D. in Computer Science

*Sept. 2017 - Jul. 2023*

ShanghaiTech University, Shanghai

Supervised by Prof. Fu Song

Mainly working on developing new techniques to

- formally verify side-channel resistance of cryptographic algorithms
- design secure and efficient masked implementations against side-channel attacks

### BSc in Computer Science

*Sept. 2013 - Jul. 2017*

China University of Mining and Technology(CUMT), Xuzhou, Jiangsu

GPA: 3.70/4.00

Thesis: Static Memory Leak Detection of Rust Programs

## RESEARCH PUBLICATION

---

### SCInfer: Refinement-based Verification of Software Countermeasures against Side-Channel Attacks

Jun Zhang, Pengfei Gao, Fu Song and Chao Wang.

*In Proceedings of the 30th International Conference on Computer Aided Verification (CAV), 2018. (CCF-A)*

$\diamond$  I was involved in designing a new semantic type inference approach for verifying masking countermeasures and a method for gradually refining the type inference system using SMT solver based analysis, to ensure the overall method is complete.

### Quantitative Verification of Masked Arithmetic Programs against Side-Channel Attacks

Pengfei Gao, Hongyi Xie, Jun Zhang, Fu Song and Taolue Chen.

*In Proceedings of the 25th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (ETAPS/TACAS), 2019. (CCF-B)*

$\diamond$  I was involved in designing a hybrid approach which integrates type system based and model-counting based approaches into a framework, and supports a sound and complete reasoning of masked arithmetic programs. In case when the masking is not effective, we provide quantitative analysis to calculate the information leakage. I designed and implemented a tool QMINFER and conducted experiments on various benchmarks.

### Verifying and Quantifying Side-Channel Resistance of Masked Software Implementations

Pengfei Gao, Jun Zhang, Fu Song and Chao Wang.

*ACM Transactions on Software Engineering and Methodology (ACM TOSEM), 2019. (CCF-A)*

$\diamond$  This is the extension of our CAV 2018 work. I was involved in designing and implementing a new algorithm to compute the quantitative masking strength (QMS) values for intermediate results that are not perfectly masked.

### Formal Verification of Masking Countermeasures for Arithmetic Programs

Pengfei Gao, Hongyi Xie, Pu Sun, Jun Zhang, Fu Song, and Taolue Chen

*IEEE Transactions on Software Engineering (IEEE TSE), 2021. (CCF-A)*

$\diamond$  This is the extension of our TACAS 2019 work. I was involved in designing a type system supporting compositional reasoning inspired from assume-guarantee framework and some domain specific heuristics which can efficiently and effectively prove masking countermeasures. I designed and implemented a

tool QMVERIF and conducted extensive experiments on various benchmarks including AES, DES and MAC-Keccak.

### **A Hybrid Approach to Formal Verification of Higher-Order Masked Arithmetic Programs** Pengfei Gao, Hongyi Xie, Fu Song and Taolue Chen.

*ACM Transactions on Software Engineering and Methodology (ACM TOSEM), 2021. (CCF-A)*

◊ In this work, I was involved in designing a sound type system and an efficient type inference algorithm for proving the security of higher-order masked implementations. We also designed a GPU-accelerated parallel algorithm to resolve potential leakages and a novel pattern-matching-based method to automatically summarize patterns of leakages, which can reduce the cost of model counting. I designed and implemented a tool HOME and conducted experiments on various benchmarks.

### **Model-based Automated Testing of JavaScript Web Applications via Longer Test Sequences**

Pengfei Gao, Yongjie Xu, Fu Song and Taolue Chen.

*Frontiers of Computer Science, 2022. (CCF-C)*

◊ I was involved in designing the first method to construct finite-state machine models to represent the behaviors of JavaScript Web applications, taking both the previously executed events and DOM event dependency into account. I also presented a new automated testing approach for generating longer event sequences of client-side JavaScript Web applications by leveraging the proposed finite-state machine models. I designed and implemented an open source tool LJS and demonstrated its efficiency and effectiveness.

## **SERVICE**

---

<b>Student Volunteer</b>	ISSTA 2019, ETAPS 2019
<b>Sub-reviewer</b>	CAV 2021, IEEE TSE

## **SKILLS**

---

**Proficient:** Static Program Analysis, Side-Channel Attacks, Compositional Reasoning, SMT-based Reasoning, Model-based Testing, Java, C++, Python, L<sup>A</sup>T<sub>E</sub>X

**Familiar:** Programming Languages and Compiler, Fuzzing, Symbolic Execution, Rust

**Knowledgeable:** OCaml, UPPAAL

## **HONOURS & SCHOLARSHIPS**

---

<b>Honours</b>	Merit Student at ShanghaiTech (2021)
	Merit Student at ShanghaiTech (2020)
	Excellent Student at ShanghaiTech (2019)
	Excellent Graduate of CUMT (2017)
	Excellent Graduation Thesis from CUMT

<b>Scholarships</b>	Baosteel Excellent Student Scholarship (2021)
	National Scholarship for Graduates (2020)
	CSC-IBM Excellent Chinese Student (2019)
	ETAPS Scholarship (2019)
	FLoC Travel Grant (2018)