Ensuring the security of systems is crucial in computer science. With different emphases, formal methods and cryptography both pursue this goal with similar rigor, methodology, and reliance on mathematics. Successful stories such as EasyCrypt [1] and the verified implementation of TLS [2] have showcased the tremendous potential in the intersection between these two fields. An in-depth understanding and synergy of cryptography and formal methods can undoubtedly unlock even greater achievements and possibilities. I aim to achieve reliability, verifiability, and confidentiality for software and systems by combining research in these two fields and focusing on two significant opportunities that have been overlooked in the past:

- Establishing **verifiability for all programs**, regardless of their source code accessibility. The lack of trust and safety guarantees in closed-source software is a significant issue. Closed-source software remains prevalent in many applications, even when it comes to critical infrastructure. My research addresses this issue by developing a privacy-preserving formal verification toolchain that allows the correctness of software to be verified, even when the code of that software is not available. So far, the toolchain has supported various fundamental formal verification tasks such as model checking [3], regular expression pattern matching [4], SAT solving [5], unsatisfiability certification (ZKUNSAT) [6], and interdomain network verification [7, 8]. Notably, ZKUNSAT demonstrates deployment readiness of privacy-preserving formal verification on real-world applications and received recognition as **a distinguished paper at CCS 2022**.
- Making **cryptography usable for all programmers**, regardless of their level of expertise. Although powerful, many cryptographic tools are hard to use and slow to run. My research utilizes techniques from automated reasoning and cryptography to design programming frameworks for the analysis, synthesis, and optimization of cryptographic implementations. My recent work, Ou [9], empowers programmers with limited knowledge of cryptography to develop zero knowledge proof (ZKP) applications effortlessly. At the same time, the Ou framework efficiently and automatically parallelizes the ZKP application, enabling its deployment across a cluster of machines. This work received a **Yale Roberts Innovation Award** and is now poised for commercialization. It has attracted huge interest from the industry and investors.

Research agenda. In my future work, I will continue pushing the frontier of the intersection of formal methods and cryptography. First, I aim to build comprehensive privacy-preserving verification tools for real-world programs even when access to the source code is restricted. My work has demonstrated both the potential of and the need for such technologies. On the one hand, such efforts will enhance expressiveness by accommodating programming languages at varying abstraction levels, from high-level languages such as C++ to low-level assembly binary. On the other hand, I will explore other privacy-enhancing tools and verification techniques, such as differential privacy and fuzzing, that improve the efficiency of the verification process by relaxing the privacy and safety guarantees. Combining these elements will yield to privacy-preserving software verification systems towards the goal of all software being verifiable and trustworthy. Second, my future work will involve extending the dedicated deployment frameworks described above to support other approaches in cryptography, such as homomorphic encryption and secure multi-party computation. I will focus on automated optimizations of the compiled code to ensure efficient usage of tools in cryptography, so that the full potential of cryptography can be unleashed in practical applications. Last, I will apply my knowledge across related domains, including hardware intellectual property protection, language-based security, network verification and optimization, and machine learning-based code generation. My ultimate aim is to lead the effort towards a future where formal methods and cryptography unite to provide enhanced security for computer systems.

References

[1] Gilles Barthe, François Dupressoir, Benjamin Grégoire, César Kunz, Benedikt Schmidt, and Pierre-Yves Strub. Easycrypt: A tutorial. *Foundations of Security Analysis and Design VII: FOSAD 2012/2013 Tutorial Lectures*, pages 146–166, 2014.

[2] Lawrence C Paulson. Inductive analysis of the internet protocol tls. *ACM Transactions on Information and System Security (TISSEC)*, 2(3):332–351, 1999.

[3] Samuel Judson, Ning Luo, Timos Antonopoulos, and Ruzica Piskac. Privacy preserving ctl model checking through oblivious graph algorithms. In *Proceedings of the 19th Workshop on Privacy in the Electronic Society*, pages 101–115, 2020.

[4] Ning Luo, Chenkai Weng, Jaspal Singh, Gefei Tan, Ruzica Piskac, and Mariana Raykova. Privacy-preserving regular expression matching using nondeterministic finite automata. *Cryptology ePrint Archive*, 2023.

[5] Ning Luo, Samuel Judson, Timos Antonopoulos, Ruzica Piskac, and Xiao Wang. ppsat: Towards two-party private sat solving. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 2983–3000, 2022.

[6] Ning Luo, Timos Antonopoulos, William R Harris, Ruzica Piskac, Eran Tromer, and Xiao Wang. Proving unsat in zero knowledge. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2203–2217, 2022.

[7] Ning Luo, Qiao Xiang, Timos Antonopoulos, Ruzica Piskac, Y Richard Yang, and Franck Le. Iveri: Privacy-preserving interdomain verification. *arXiv preprint arXiv:2202.02729*, 2022.

[8] Yichao Cheng, Ning Luo, Jingxuan Zhang, Timos Antonopoulos, Ruzica Piskac, and Qiao Xiang. Looking for the maximum independent set: a new perspective on the stable path problem. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2021.

[9] Yuyang Sang, Ning Luo, Samuel Judson, Ben Chaimberg, Timos Antonopoulos, Xiao Wang, Ruzica Piskac, and Zhong Shao. Ou: Automating the parallelization of zero-knowledge protocols. *Proceedings of the 2032 ACM SIGSAC Conference on Computer and Communications Security*, 2023.