As our reliance on digital techniques grows, ensuring the reliability and security of software becomes paramount for the economy, critical infrastructure, and data privacy. My research is pioneering the synergies between cryptography and formal methods to guarantee security and reliability in software on a broader scale. Classical formal methods of the source code can be performed to check if some software is reliable and secure. However, these approaches are unfeasible when the source code is inaccessible. The enduring debate about the level of trust that should be placed in closed-source software persists. To address this challenge, my research **initiates the research on formal verification of software when the source code is private and inaccessible**. Besides the theoretical groundwork, the attainment of reliable and secure software relies on a substantial cybersecurity workforce to deploy the theoretical solutions in practice. Human efforts alone may not be sufficient. My research is **at the forefront of developing computer-aided frameworks for secure and optimal deployments**.

- **Ensure secure software regardless of source code accessibility.** Building a software market that promotes security-enhanced practices and that preserves innovation is one objective in the National Cybersecurity Strategy.[1] Many software products, particularly those with cutting-edge innovation, offer restricted access to their source code, making security assessments challenging. My research is at the frontier of software verification systems that can ensure the correctness, robustness, and safety of programs without exposing the source code. My work has covered a wide span of source code representations, including Boolean formulae [3], configuration files [1, 6], regular languages [5], and transition systems [2].
- **Empower cybersecurity workforce via automation.** The global shortage of cybersecurity experts is a growing concern. Projects executed by non-expert programmers pose inherent risks due to their lack of domain knowledge. My research complements the cybersecurity workforce by providing accessible computer-aided programming frameworks for developers with varying expertise levels. One example is my work, Ou [7], which concentrates on the secure implementation of zero-knowledge proof (ZKP) protocols. It pioneers the research into automating and optimizing the parallelization and distribution of ZKP. Another challenge is identifying and resolving conflicts among untrusted collaborators. These conflicts often emerge from different parties in constraints imposed by regulations, obligations, or policies. To further boost workforce efficiency and effectiveness, my work, ppSAT [4], initiates the design of systems to identify and resolve conflicts among untrusted stakeholders.

Verification for Closed-Source Software

Closed-source software in digital marketplaces fuels innovation, but it also raises security risks. Security verification could allay these concerns, but appears at first glance to be an impossible task because of the restricted access to the source code. Demanding universal open-sourcing is impractical, as evidenced by the industry controversy around the US Defense Innovation Boards's 2019 recommendation[2] that the Department of Defense require source code access from vendors. With the growing demand for digital markets that preserve security and innovation, my research aims to achieve verification while protecting source code. In particular, my work provides a privacy-preserving formal verification toolchain that allows software verification even when the code and design of that software are protected and hidden. The toolchain progresses from foundational algorithms to higher-level abstractions, starting with Boolean reasoning.

My work, ZKUNSAT [3] enables the verification of unsatisfiability of Boolean formulas while ensuring the confidentiality of the formulas through ZKP. As **the first practical ZKP for co-NP problems**, it can verify the safety of Windows audio drivers within five minutes. In software verification, guaranteeing that a program complies with a required property can be accomplished by the developer transforming the program and the property into a formula. The unsatisfiability of this formula confirms that the program adheres to the

[1] White House. National Cybersecurity Strategy (2023). Retrieved from https://www.whitehouse.gov/wp-content/uploads/2023/03/National-Cybersecurity-Strategy-2023.pdf

[2] Department of Defense. DIB Clarification of Source Code (2019). Retrieved from https://media.defense.gov/2019/Oct/31/2002204223/-1/-1/0/DIB_CLARIFICATION_OF_SOURCE_CODE.PDF

desired property. ZKPs are cryptographic tools that enable one party (the prover) to substantiate a claim to another party (the verifier) without disclosing confidential evidence. Applying ZKPs to unsatisfiability for verification formulas allows the developer (as the prover) to establish the correctness of the program to another party (the verifier) while ensuring the formula remains private, which consequently also safeguards the confidentiality of their program. To avoid significant overhead from applying ZKP, I developed a specialized encoding for formulae and refutation proofs in ZKUNSAT. Such encoding, unexplored in traditional unsatisfiability verification settings, has outstanding performance when integrated with ZKP. This work received the **Distinguished Paper Award** at the ACM CCS 2022, recognizing its significant contribution.

ZKSMT, my recent work, takes another step towards verification on real-world private programs by enabling proof of the unsatisfiability of SMT formulae in ZKP. At the heart of many software verification techniques, SMT formulae extend from Boolean formulae to include first-order theories, such as those related to integers and arrays. The existing tools for verifying the unsatisfiability of SMT formulae in ZKP lack scalability, even when applied to small examples. For instance, it can take up to two hours to use a cutting-edge ZKP toolchain to process a brief benchmark of just six operations. To address this, my work introduces the ZKSMT system. This system is designed to be compact, general, and compatible with common ZKP optimizations all at once. It include a novel format for refutation proofs of unsatisfiable SMT formulas. Meanwhile, the system offers modular support for integrating new theories (such as cleartext SMT solvers) and provides the flexibility to add customized protocols in ZKP for various theories. As a result, ZKSMT meets the goals of compactness, generality and efficiency. We benchmarked our system against the Wisconsin Safety Analyzer benchmark suite. The results show that ZKSMT achieves a speed-up of more than three orders of magnitude compared to the prior state-of-the-art system.

The impact of privacy-preserving verification reaches beyond software alone. GreedMIS [1] addresses the stable path problem (SPP) for interdomain network verification, all while preserve the privacy of routing information for each autonomous system (AS). ASes can collaboratively compute a solution to SPP that ensures the convergence of their interdomain policies. However, this needs the exchange of private and sensitive routing information and policies GreedMIS provides tools that allow untrusted ASes to determine stable path assignments without compromising the confidentiality of their routing preferences.

Besides the above, the toolchain in my work also supports other fundamental verification tasks covering model, checking [2], regular expressions [5], and configuration files [6].

Strengthening the Cybersecurity Workforce via Automation

Given the shortage of cybersecurity experts, it is impractical to anticipate that every detail of security-related projects can be overseen by qualified professionals. Enhancing the cybersecurity workforce with automation enables the available experts to increase their capacity without sacrificing quality, and can sometimes provide a reliable and cost-effective alternative to domain experts altogether. My research complements human efforts by providing automated programming frameworks for developers at all levels of domain expertise.

My work, Ou [7], focuses explicitly on deploying ZKP protocols. Despite ZKP's impressive capabilities, they impose a notable resource overhead. For example, state-of-the-art ZKP protocols often require substantial memory resources, and inherent bottlenecks have been identified. Existing deployment frameworks for ZKP guarantee secure and accurate implementations through constrained interfaces. Nevertheless, these constraints curtail optimization opportunities, limiting flexibility and causing the frameworks to generate proofs with suboptimal efficiency. Ou enables programmers with various levels of cryptographic expertise to develop ZKP applications without difficulty. The frontend language of Ou is designed to resemble C++ and ensures privacy correctness through its robust type system. It significantly reduces the level of cryptographic expertise needed by programmers to deploy efficient ZKP protocols. Ou's frontend language further stands out due to its great expressiveness, thanks to a wide array of annotations. Using these annotations, cryptography experts can introduce optimizations when their expertise enables it, such as employing extended witnesses or random challenges. When used, however, such annotations introduce non-determinism into the deployed ZKPs. To ensure secure and correct development, Ou sandboxes the non-determinism to quarantine the effect and guarantee privacy and correctness. In addition to its programming language design, Ou is the first programming framework for ZKP that provides fully automated and optimal parallelization. This is achieved by formulating program parallelization as integer linear programming problems. The correctness of the output parallelized ZKP is then certified through our formally defined semantics of the ZKP language. This work received a **Yale Roberts Innovation Award** and is now poised for commercialization.

Collaboration within the private sector is crucial for enhancing the efficiency of the national and even global cybersecurity workforce. However, such widespread collaboration presents notable challenges because many stakeholders uphold their individual set of policies, interests, constraints, and strategies that should be kept confidential. My research, ppSAT [4], pioneers solutions to the challenge of human-to-human collaboration through machine-to-machine protocols. It combines recent advancements in oblivious algorithm design with classic techniques for SAT solving and secure two-party computation to develop a privacy-preserving solver for Boolean satisfiability. As such, ppSAT enables mutually distrustful parties to decide the satisfiability of the conjunction of their individual Boolean formulae without revealing the contents. Many policies, interests, constraints, and strategies can be effectively expressed as Boolean formulae. ppSAT therefore can provide a foundation for consensus-building, policy refinement, and conflict resolution in untrusted settings. To ensure practicality, ppSAT introduces specialized data structures and heuristics related to but distinct from those of cleartext solvers. The data structures and heuristics employed in modern SAT solvers — often critical to their success — are heavily dependent on the specific inputs and not well-suited for privacy-preserving scenarios. ppSAT's data structures and heuristics are carefully engineered to incorporate the strengths of both worlds: they adhere to the foundational principles of classical SAT solving while making preserving privacy practical. The evaluation results of ppSAT demonstrate the feasibility of privacy-preserving Boolean reasoning and highlight the potential for further advancements in privacy-preserving SAT solving. More than just the research community have also shown interest in this work: we're partnering with Google to integrate ppSAT within their toolchain.

Ongoing and Short-Term Future Work

Expressiveness, composability, and scalability. I will construct comprehensive privacy-preserving verification toolchains that can verify programs across various abstraction levels. Software verification success depends on carefully designed tools for various levels of program abstractions and their composition. I aim to create tools of privacy-preserving verification for 1) program representation spanning high-level languages like C++ to low-level assembly binaries; and 2) refinement relations that ensure program representations at different levels accurately reflect the behaviors of the programs of interest. My future work will also improve the scalability of privacy-preserving verification tools by introducing new data structures and heuristics that are efficient under privacy constraints.

Efficient ZKP for PSPACE. Recent developments in ZKP research have primarily focused on problems in NP. While ZKUNSAT unleashes the potential for practical ZKPs for co-NP problems, ZKPs for PSPACE still remain as theoretical answers. However, plenty of challenging and exciting problems, such as regular expression equivalence, lie beyond the scope of NP but are of great practical interest. Making ZKPs for PSPACE as practical as those in NP poses presents a challenging and attractive task. I am leading the project to extend the practical applicability of ZKPs to PSPACE. A significant obstacle in achieving this goal lies in the absence of efficient provers capable of finding certificates or witnesses for problems within these classes. Leveraging recent advancements in formal language theory and automated reasoning techniques, my objective is to surmount this obstacle and develop practical protocols.

Synthesizing cryptographic designs and deployments. The development of efficient implementation alternatives that are functionally equivalent to slower programs has been the subject of extensive research for a significant period. However, they are not widely adopted when it comes to cryptography. In my future research, I will focus on synthesizing efficient cryptographic protocol implementations, with the initial step being centered around ZKP. There are numerous possibilities for optimization in ZKP due to the knowledge of the prover and the context of verification. The former can provide more hints in the computation, and the latter can take a broader perspective on functional equivalence. I am now spearheading a NSF-awarded project to achieve this. Additionally, choosing the most appropriate instantiation for a particular functionality is vital for optimizing the overall system's performance. However, optimal selections become particularly challenging under the increased complexity in real-world systems consisting of a multitude of functionalities. These functionalities are interlinked, and their performances exert a mutual influence. In the future, I will develop automated tools to offer optimal strategies for selecting and scheduling cryptographic instantiations within complex software systems.

Privacy-aware conflict resolution and policy refinement. To establish collaboration and agreement, the identification of conflicts within the constraints of multiple parties, achieved by ppSAT [4], is insufficient. My objective is to design privacy-preserving protocols for addressing these conflicts by refining each party's constraints that reflect their policies or strategies. In real-life scenarios, these constraints are often multifaceted, covering a range of different metrics. Some may be inflexible and firmly established, while others demand a heightened level of confidentiality, and so forth. Within this intricate, multidimensional framework, finding the most effective resolution and refinement under privacy-preserving constraints represents a significant challenge. Designing and deploying protocols to address this holds both theoretical and practical interests. I am collaborating with Google to make progress in this endeavor.

Long-Term Future Work

Automated reasoning in the cloud. My research focuses on facilitating cloud-automated reasoning. When deployed as a cloud-based service, automated reasoning holds the potential for substantial benefits, although it is not without its challenges. One promising avenue is the ability of cloud servers to efficiently reuse proven theorems, which can be stored in a database. This has the potential to be particularly resource-saving, given that many theorems employed in practical applications exhibit many similar patterns. Conversely, such a service would face hard security problems. Detecting denial-of-service attacks would be a complex task for cloud servers, given the inherently time-consuming nature of automated reasoning. Recognizing malicious queries calls for the development of innovative learning techniques. In the future, I will focus on addressing these challenges and advancing the transformation of automated reasoning into a cloud-based service.

Accountability and liability of software. Our lives and society are increasingly impacted by the decisions made by software. Despite efforts to assess and verify the safety, correctness, and fairness of such algorithms, the inherent hardness and undecidability of their verification cannot completely eliminate the risk of severe failures. When these algorithms inevitably cause harm, accountability and liability become critically important issues. I will investigate automated-reasoning-based techniques to aid investigators in investigating software accountability and liability. My work aims to offer tools for software users and enable them to assess whether their settings align with the policies and regulations of the systems they utilize. To accomplish this, effective interdisciplinary collaboration is crucial, bridging the gap between computer science, legal philosophy and practice, and public policymaking.

Testing and verifying non-deterministic systems in the fields. I am interested in explicitly addressing non-determinism within various contexts. Non-deterministic systems are found in many fields, such as machine learning, concurrent systems, distributed computation, and cyber-physical systems. Challenges in testing and verification for these fields arise from modeling and reasoning the non-determinism. For example, testing is challenging for cyber-physical systems because environmental inputs are out of testers' control and sometimes not repeatable. In Ou, we use specially designed sandbox techniques to isolate the effects of non-deterministic behaviors in the ZKPs. In the future, I will explore approaches to reasoning about non-determinism in other fields, such as quantum computing and autonomous vehicles.

References

[1] Y. Cheng, N. Luo, J. Zhang, T. Antonopoulos, R. Piskac, and Q. 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.

[2] S. Judson, N. Luo, T. Antonopoulos, and R. 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.

[3] N. Luo, T. Antonopoulos, W. R. Harris, R. Piskac, E. Tromer, and X. Wang. Proving UNSAT in zero knowledge. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2203–2217, 2022.

[4] N. Luo, S. Judson, T. Antonopoulos, R. Piskac, and X. Wang. ppSAT: Towards two-party private sat solving. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 2983–3000, 2022.

[5] N. Luo, C. Weng, J. Singh, G. Tan, R. Piskac, and M. Raykova. Privacy-preserving regular expression matching using nondeterministic finite automata. *Cryptology ePrint Archive*, 2023.

[6] N. Luo, Q. Xiang, T. Antonopoulos, R. Piskac, Y. R. Yang, and F. Le. Iveri: Privacy-preserving interdomain verification. *arXiv preprint arXiv:2202.02729*, 2022.

[7] Y. Sang, N. Luo, S. Judson, B. Chaimberg, T. Antonopoulos, X. Wang, R. Piskac, and Z. Shao. Ou: Automating the parallelization of zero-knowledge protocols. *Proceedings of the 2032 ACM SIGSAC Conference on Computer and Communications Security*, 2023.