Master's Internship and PhD Proposal
Dolev-Yao-Model-Guided Fuzzing of E-Voting Protocols

Duration: 5 to 9 months for a Master's internship; 3 to 4 years for a PhD; starting in 2025 or 2026

Where: At Inria in Nancy, France. In the research team PESTO at LORIA. Nancy is a vibrant university city in northeastern France with excellent research facilities and quality of life.

Name and email address of the advisors
Lucca Hirschi, `lucca.hirschi@inria.fr` & Alexandre Debant, `alexandre.debant@inria.fr`

Funding
The internship or PhD is supported by the ProtoFuzz ANR (program managed by the French National Research Agency under grant agreement No. ANR-22-CE48-0017). There will be a stipend for the Master's internship. The salary for the PhD candidate is in line with French national standards: monthly gross salary from 2 082 € to 2 190 €.

TL;DR. We are seeking a Master's intern or a PhD candidate to join our puffin [1] team and design Dolev-Yao (DY) model-guided fuzzing techniques [2] amenable to e-voting protocols.

Context. Critical and widely used cryptographic protocols have repeatedly been found to contain flaws in their design and their implementation. A prominent class of such vulnerabilities is logical attacks, e.g., attacks that exploit flawed protocol logic. Automated formal verification methods, based on the Dolev-Yao (DY) attacker, formally define and excel at finding such flaws, but operate only on abstract specification models. Fully automated verification of existing protocol implementations is still out of reach today. This leaves open whether such implementations are secure.

On the opposite side of the spectrum, *fuzz testing*, developed since the 90s, is now the gold standard for testing security software and is used at scale by the largest software companies. However, even if a protocol implementation is tested against memory-related vulnerabilities using state-of-the-art fuzzers, the whole class of *implementation-level logical attacks* remains out of scope. Unfortunately, this blind spot hides numerous attacks, notably recent logical attacks on widely used TLS implementations introduced by implementation bugs.

We have recently addressed this challenge by proposing a novel and effective technique that we called DY model-guided fuzzing [2], which captures logical attacks against protocol implementations. The main idea is to consider as possible test cases the set of abstract DY executions of the formal DY attacker, and use a novel *mutation-based fuzzer* to explore this set. This approach enables reasoning at a more structural and security-related level of messages represented as *formal terms* (e.g., decrypt a message and re-encrypt it with a different key) as opposed to random bit-level modifications that are much less likely to produce relevant logical adversarial behaviors. We have implemented a full-fledged and modular DY protocol fuzzer puffin [1]. We have demonstrated its effectiveness by fuzzing four popular TLS implementations, resulting in the discovery of several novel vulnerabilities, RFC non-compliance errors, and bugs[1].

This recent work has opened up various exciting new research questions we would like to explore within the puffin team; *i.e.,* Lucca Hirschi (Inria researcher), Steve Kremer (Inria senior researcher), Tom Gouville (PhD student), and Olivier Demangeon (Inria research engineer working on puffin).

Research Direction. Depending on the candidate's expertise and interests, we will collaboratively choose among several research directions we would like to explore. One promising direction is to explore the application of DY fuzzing to e-voting protocols. Such protocols are critical pieces of software which have not received much attention from the fuzzing community so far. DY fuzzing seems to be ideally placed to address the unique challenges posed by e-voting systems. The protocol and implementation we envision exploring first is that of SwissPost [4], which is often used in Switzerland. Such a protocol and implementation constitutes an interesting target for several reasons: they are based on many different *micro-services* run by different actors and they aim to achieve extremely strong security properties under strong threat models. We also have a strong expertise in e-voting protocols in our team, notably on SwissPost [3].

[1] Notably CVE-2022-42905 (critical severity), CVE-2022-39173 and CVE-2022-38152 (medium severity).

Objectives. First of all, the candidate will get familiar with formal DY models, fuzzing [2], as well as with the existing code base of SwissPost [4].

There are several challenges to adapting DY fuzzing techniques to e-voting and to SwissPost. First, e-voting protocols differ from mere internet protocols in that they involve complex workflows and interactions between different actors, implemented as different micro-services in SwissPost. Therefore, the intern will have to design a way to fuzz such complex workflows in order to animate *e-voting ceremonies and elections*. Second, the security properties of e-voting protocols are different and more complex than those of authentication protocols such as TLS. Therefore, the candidate will have to design *objective oracles* capable of detecting violations of e-voting security properties. Finally, SwissPost is supposed to resist strong threat models, including insider threats and collusions between different actors. Therefore, the candidate will have to design dynamic actor compromise within the DY fuzzer.

The precise direction this project will take shall be agreed upon with the candidate at the beginning of the project. The candidate will benefit from dedicated engineering support from our research engineer working on the project, while also being expected to contribute independently to the software development efforts. Should we find any vulnerability, we would follow standard and ethical responsible disclosure practices.

Internship candidates interested in pursuing an academic career may also consider continuing towards a PhD, as we have additional funding available for promising candidates.

Required Qualifications and Expectations. We expect mathematical maturity and basic knowledge in logic and theoretical computer science. Knowledge in security and cryptography is not mandatory but is definitely a plus. For the implementation work, proficiency in Rust programming is also a plus.

How to Apply. Interested candidates should send the following documents to both advisors:

- a detailed CV,
- a cover letter explaining their motivation and relevant experience,
- contact information for 1-3 academic and teachers references,

References

[1] puffin project. Main webpage: https://tlspuffin.github.io/. Source code: https://github.com/tlspuffin/tlspuffin. Funding: https://project.inria.fr/protofuzz/.

[2] Max Ammann, Lucca Hirschi, and Steve Kremer. DY fuzzing: formal dolev-yao models meet cryptographic protocol fuzz testing. In *Symposium on Security and Privacy*. IEEE, 2024.

[3] Véronique Cortier, Alexandre Debant, Olivier Esseiva, Pierrick Gaudry, Audhild Høgåsen, and Chiara Spadafora. A practical and fully distributed e-voting protocol for the swiss context. *Cryptology ePrint Archive*, 2025.

[4] Swiss Post. Swiss Post Voting System: System Specification. Version 1.5.0. Technical report, 2025.