Mukesh Tiwari
Cambridge, United Kingdom
+447824648138
mt883@cam.ac.uk

To
The Hiring Committee
The University of Luxembourg, Luxembourg

Application for the post of Assistant Professor in Software Engineering

Dear Hiring Committee,

My name is Mukesh Tiwari and I am senior research associate at the University of Cambridge, Cambridge, UK. I am writing to apply for the job **Assistant Professor in in Software Engineering** . I have an extensive experience in formal verification (Coq theorem prover) and security research (Cryptography), and I find the University of Luxembourg will be a perfect place to continue my research in formal verification and expand my horizons in other areas of software engineering, including AI for software development, program synthesis, etc.

Nowadays, we are relying more and more on software programs for decision making. For example, many government entities are making policies based on the output of a software program. However, if the software program contains bugs, then it may produce a wrong output. Therefore, the government entity can lose its reputation. In addition, it can also hamper the trust of members of general public in the government decision-making processes. Therefore, it is more imperative than ever that we formally verify these software programs and develop tools to automate the formal verification process to make the government decision-making more trustworthy.

In my PhD, I have formally verified the Schulze vote counting method. I have addressed ballot privacy by using homomorphic encryption, and verifiability by means of producing an independently verifiable scrutiny sheet (certificate), the validity of which can be independently substantiated, that witnesses the correctness of the execution of an election. As a research associate at the University of Melbourne, I did acquire hands-on knowledge of separation logic and information flow security. I have spearheaded two projects: (i) a formally verified auction server and (ii) a formally verified location server. Both implementations were proven memory safe (using separation logic) and free from information leaks (applying information flow security), using the SecCSL tool. In addition, I also explored side channel attacks that might arose due to secret-dependent branching of the code running inside Intel SGX (Software Guard Extensions) enclave. Currently, in Cambridge, I am working on formal verification of graph algorithms on semirings. In this setting, a formally verified graph algorithm can compute different things depending on the concrete structure of semiring, e.g., the same algorithm can compute shortest path, longest paths, data flow of imperative programs, and many more for an appropriate choice of semiring.

My long term goal is to make formal verification accessible for all kind of software programs, and my current goal is to formally verify all the software programs used in public domain, especially used in voting. I look forward to hearing from you. Let me know if you have any questions.[1]

Your Sincerly,

Mukesh Tiwari

My research had been severly impacted by Melbourne lockdown and therefore I was not very productive in year 2020 and 2021.