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

To
Prof. Vanessa Teague,
Australian National University, Canberra

Application for the Research Fellow - Election Security

I am writing to apply for the **Research Fellow - Election Security** job. I have extensive research experience in theorem proving (Coq theorem prover), (election) software security, cryptography, and social choice theory. My research is highly relevant to a democracy that conducts an election using software programs. I have developed many formally verified solutions that address the problems, pointed by researchers, of e-voting software programs. For example, I have developed a formally verified independently verifiable group generator to bootstrap elections, a formally verified scrutiny sheet checker for Swiss elections, replacing the SwissPost Java cryptographic implementation by a formally verified Coq implementation and proofs, formally verified the Schulze method, etc. I have a PhD from the Australian National University, Canberra, Australia, and currently, I am working as a senior Research Fellow at the University of Cambridge. Before moving to Cambridge, I was a research fellow at the University of Melbourne, Australia.

The goal of my PhD was to bring three important ingredients, correctness, privacy, and (universal) verifiability, of a paper ballot election to an electronic setting (electronic voting). I demonstrated: (i) correctness by implementing and proving the correctness of a vote-counting algorithm, the Schulze method, in the Coq theorem prover, (ii) privacy by using homomorphic encryption to encrypt the ballots and computed the winner by combining all the (encrypted) ballots, and (iii) verifiability by means of various zero-knowledge-proofs. My research has been published in Interactive Theorem Proving (ITP), Computer and Communications Security (CCS), Electronic Voting (EVote), International Conference on Cryptology in India (IndoCrypt), and some (finished) works have been submitted to USENIX and SIGCOMM. All my research work has been formalised in the Coq theorem prover.

Many countries, including Australia, are using software programs to count ballots, For example, Australian Election Commission scans all the ballots of the senate election and uses a (closed source) software program to produce the final tally. However, it is highly undesirable because software programs generally contain bugs and therefore, they may produce a wrong winner than the the intended one. For example, Rina Mercuri, a candidate in the council of Griffith, narrowly missed out on a seat because of a closed source software bug (opaqueness), shown by Andrew Conway, Michelle Blom, Lee Naish, and Vanessa Teague in the paper *An analysis of New South Wales electronic vote counting*. Unfortunately, it is not an isolated event of opaqueness and lack of transparency. In fact, opaqueness was one of the major reason for Bundesverfassungsgericht, the German Constitutional Court, to rule the usage of electronic voting machines unconstitutional. However, it did not completely rule out the usage of electronic voting machines as long as the outcome of an election is verifiable, i.e., it is possible for the citizen to check the essential steps in the election act and in the ascertainment of the results reliably and without special expert knowledge (verifiability). Thefore, to improve the current situation of election security, I will also seek external funding (ARC DECRA) to continue the election security research.

Regarding teaching, I have had 3 years of teaching at a small technical university, International Institute of Information Technology, Bhubaneswar, India and tutoring experience at the Australian National University. At ANU, I feel qualified to teach most of the computer science courses because of my background in computer science, but my natural preference is the courses close to my research area, e.g., theorem proving, cryptography, logic and its applications in computer science, discrete mathematics, algorithms and data structures, introductory programming course (C/C++/Java/Python/Haskell/OCaml), foundation of computing, etc. In addition, I would also like to design a course to teach various voting methods used around the world and their advantages and disadvantages from a social choice theory perspective. Apart from these topics, I will be more than happy to teach other courses, given enough preparation time, at introductory and intermediate levels, e.g., type theory,

theory of programming languages, networking, operating systems, databases, etc.

In my current project *Combinators for Algebraic Structures (CAS)*, I am formalising various graph algorithms on *semiring* algebraic structures and combinators (functions) to combine two, or more, algebraic structures. In this work, I am developing a mathematical correct-by-construction framework, in Coq theorem prover, based on theory of generalised path-finding algebra. In our framework, depending on concrete instantiation of semiring operators, the same algorithm can compute shortest path, longest paths, widest paths, multi-objective optimisation, data flow of imperative programs, and many more. In fact, the Schulze method is one instance of our framework. However, the current CAS implementation is highly focused towards networking protocols, so as a future work I will focus on adding more algorithms in CAS related to multi-objective optimisation, data flow analysis of imperative programs, and voting.

As a research associate at the University of Melbourne, I have spearheaded three projects: (i) a formally verified auction server, (ii) a formally verified location server, and (iii) a formally verified machine learning algorithm that is resistant to side-channel attacks and can run inside the Intel SGX (Software Guard Extensions) enclave for learning on sensitive data. All three implementations have been proven memory safe (using separation logic) and free from information leaks using the SecureC tool, a tool developed at the University of Melbourne.

I look forward to hearing from you. Please let me know if you have any questions.

Your Sincerly,

Mukesh Tiwari