Research Statement

Mukesh Tiwari

My work aims to build **correct software programs** using the Coq theorem prover. I focus on formal verification of software programs used in elections, cryptography, networking, and computational social choice theory. Numerous critical decisions, e.g., producing the winner of an election by an election commission, are taken 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 (election commission) can lose its reputation.

My PhD research was focused on verifying electronic voting, specifically vote-counting schemes, in the Coq theorem prover. The goal was to bring three important ingredients, correctness, privacy, and (universal) verifiability, of a paper ballot election to an electronic setting (electronic voting). In a paper ballot election, correctness is ensured by scrutineers, and privacy and verifiability come for free because of secret paper ballots. However, achieving these three desirable properties are difficult in electronic voting because software programs, used in various stages of an election, work in a opaque (blackbox) manner.

My long-term aim is to make formal verification accessible and ubiquitous in software development, specifically for the software programs deployed in public domain that affect common people. My expertise in **Theorem Proving, Cryptography, and Election Security** gives me an unique perspective to solve challenging problems that matter to many democracies and its citizens. In future, I will:

- focus on formally verified cryptographic primitives used in electronic voting, Internet of Things (IoT), and blockchain, e.g., sigma protocols (zero-knowledge-proof), verifiable (shuffling) mix-networks, multi-party computations, secret sharing, secure communication, zk-snark, etc.
- focus on developing formally verified (electronic) voting software (components) programs in Coq theorem prover. The rationale is that once we have formally verified components, anyone –election commission or members of general public– can use them to conduct elections, referendums, and verify elections' outcome.
- focus on formally verified decentralised peer-to-peer technical solution, inspired by [1, 2, 3], in Coq theorem prover which will help whistleblowers in leaking documents and exposing corruption without revealing their identity.
- focus on formally verified combinators for algebraic structure (CAS) in Coq theorem prover. Currently, CAS formalisation is highly focused on networking protocols, but it can be adapted for other areas, e.g., optimisation, clustering, algebraic program analysis, etc. In this setting, an algorithm can compute different values 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 [4] for an appropriate semiring.

References

[1] Joseph K Liu, Victor K Wei, and Duncan S Wong. Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups. In *Australasian Conference on Information Security and Privacy*, pages 325–335. Springer, 2004.

[2] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. *Freenet: A Distributed Anonymous Information Storage and Retrieval System*, pages 46–66. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[3] zzz (Pseudonym) and Lars Schimmer. Peer Profiling and Selection in the I2P Anonymous Network. In *PetCon 2009.1*, pages 59–70, 2009.

[4] Michel Gondran and Michel Minoux. *Graphs, Dioids and Semirings: New Models and Algorithms*, volume 41. Springer Science & Business Media, 2008.