Mukesh Tiwari

Education

2016–2020 **PhD, Computer Science**, *Australian National University*, Canberra, Australia

2004–2009 **Integrated Post Graduate**, *Indian Institute of Information Technology & Management*, Gwalior, India

PhD thesis

Title *Formally Verified Verifiable Electronic Voting Scheme*

Supervisor Dirk Pattinson

Description We focussed on three main challenges posed by electronic voting: correctness, privacy, and verifiability. We addressed correctness by using a theorem prover to implement a vote-counting algorithm, privacy by using homomorphic encryption, and verifiability by generating a independently checkable scrutiny sheet. Our work had been formalised in the Coq theorem prover.

2021- **Senior Research Fellow**, *University of Cambridge*, Cambridge, United Kingdom
I am working on formalising network protocols framework in Coq theorem prover. The goal is to develop a mathematical proven correct framework so that a protocol designer can assess the properties of their protocols using my framework

2020-21 **Research Fellow**, *University of Melbourne*, Melbourne, Australia
I worked with Toby Murray on *Security Concurrent Separation Logic*. The aim was to mathematically reason about memory safety and information flow property of concurrent programs written in C.

2013–2015 **Lecturer**, *International Institute of Information Technology*, Bhubaneswar, India
This role was primarily teaching-focussed, and the courses I taught were *C programming, Java Programming, Compiler Design* and *Cryptography*. In addition, every year I supervised two master's students in their final year project.

2012–2013 **Haskell Developer**, *Parallel Scientific*, Colorado, USA
In this role, my primary job was research and prototype high performance software programs, mainly linear algebra algorithms written in Haskell.

2009–2012 **Technical Assistant**, *Government of India*, Kolkata, India
I worked as a developer for automating the day-to-day job, including enforcing the security policies of the organisation.

2008–2008 **Summer Intern**, *Arcelor-Mittal, Research & Development Technological Centre*, Avilés, Spain
I worked on formalising many business requirements into a linear programming problem and wrote a custom interface that interacted with Arcelor-Mittal's in-house linear programming solver.

Skills

Coding Coq, Haskell, OCaml, Lean, Python, C

+44-7824648138
mt883@cam.ac.uk, mukesh.tiwari@anu.edu.au, mukeshtiwari.iiitm@gmail.com
mukeshtiwari.github.io/ • *mukesh-tiwari-3609486/*
mukeshtiwari

Language Hindi, English

Awards

HDR Fee Remission Merit Scholarship

ANU PhD Scholarship (International)

Full Scholarship to attend DeepSpec Summer School 2018, Princeton University

Travel Scholarship to attend Marktoberdorf Summer School 2019 (I could not attend it because I did not get the visa for Germany)

Conference

[1] Dirk Pattinson and Mukesh Tiwari. Schulze Voting as Evidence Carrying Computation. In Mauricio Ayala-Rincón and César A, Muñoz, editors, *Interactive Theorem Proving*, pages 410–426. Cham, 2017. Springer International Publishing. `https://github.com/mukeshtiwari/formalized-voting/blob/master/paper-code` (lead developer, project duration: 1 year)

[2] Lyria Bennett Moses, Rajeev Goré, Ron Levy, Dirk Pattinson, and Mukesh Tiwari. No More Excuses: Automated Synthesis of Practical and Verifiable Vote-Counting Programs for Complex Voting Schemes. In Robert Krimmer, Melanie Volkamer, Nadja Braun Binder, Norbert Kersting, Olivier Pereira, and Carsten Schürmann, editors, Electronic Voting, pages 66–83, Cham, 2017. Springer International Publishing. `https://github.com/mukeshtiwari/formalized-voting/tree/master/SchulzeOCaml` (lead developer, project duration: 6 months)

[3] Thomas Haines, Dirk Pattinson, and Mukesh Tiwari. Verifiable Homomorphic Tallying for the Schulze Vote Counting Scheme, In Verified Software: Theories, Tools, and Experiments. Springer, 2019. `https://github.com/mukeshtiwari/EncryptionSchulze/tree/master/code/Workingcode` (lead developer, project duration: 2 years)

[4] Milad K. Ghale, Rajeev Goré, Dirk Pattinson, and Mukesh Tiwari. Modular Formalisation and Verification of STV Algorithms. In Robert Krimmer, Melanie Volkamer, Véronique Cortier, Rajeev Goré, Manik Hapsara, Uwe Serdültt, and David Duenas-Cid, editors, Electronic Voting, pages 51–66, Cham, 2018. Springer International Publishing. `https://github.com/mukeshtiwari/Modular-STVCalculi`. (co-developer with Milad K. Ghale. I proved some of the cricital theorems, required for code extraction)

[5] Thomas Haines, Rajeev Goré, and Mukesh Tiwari. Verified Verifiers for Verifying Elections. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS '19, page 685–702, New York, NY, USA, 2019. Association for Computing Machinery. `https://github.com/mukeshtiwari/secure-e-voting-with-coq`. (co-developer with Thomas Haines, I worked on efficient finite Field arithmetic, required for efficient zero-knowledge-proof validation of well-formedness of a ballot)

+44-7824648138

mt883@cam.ac.uk, mukesh.tiwari@anu.edu.au, mukeshtiwari.iiitm@gmail.com

mukeshtiwari.github.io/ • *mukesh-tiwari-3609486/*

mukeshtiwari

[6] Nadim Kobeissi, Georgio Nicolas, and Mukesh Tiwari. Verifpal: Cryptographic Protocol Analysis for the Real World. In Karthikeyan Bhargavan, Elisabeth Oswald, and Manoj Prabhakaran, editors, Progress in Cryptology - INDOCRYPT 2020, pages 151–202, Cham, 2020. Springer International Publishing. (co-developer with Georgio Nicolas. I worked on proofs related to Verifpal model in Coq)

[7] Mukesh Tiwari, Karm V. Arya, Rahul Choudhari, and Kumar S. Choudhary. Designing Intrusion Detection to Detect Black Hole and Selective Forwarding Attack in WSN Based on Local Information. In 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology, pages 824–828, Nov 2009. (lead developer)

[8] Rahul Choudhari, Karm V. Arya, Mukesh Tiwari, and Kumar S. Choudhary. Performance Evaluation of SCTP-Sec: A Secure SCTP Mechanism. In 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology, pages 1111–1116, Nov 2009. (co-developer with Rahul Choudhari)

Workshop

[1] Mukesh Tiwari and Dirk Pattinson. Machine Checked Properties of the Schulze Method. 7th Workshop on Hot Issues in Security Principles and Trust 2021. `https://github.com/mukeshtiwari/HotSpot2021/blob/main/formal_properties_schulze.pdf`.

[2] Mukesh Tiwari. Towards Leakage-Resistant Machine Learning in Trusted Execution Environments. Program Analysis and Verification on Trusted Platforms (PAVeTrust) Workshop 2021. `https://github.com/mukeshtiwari/IFMachine/blob/main/Information_Flow_Secure_Gradient_Descent.pdf`.

Work in Progress

[1] Verified Secure Declassification for Concurrent Applications. In this work, we develop a formal model of leaking sensitive data (joint work with Toby Murray, Gidon Ernst, and David Naumann. In this work, I formally verified the case studies, location-server and aution-server, in SecureC). (submitted).

[2] Formally Verified Verifiable Group Generators. In this work, we develop a formally verified algorithm that can be used to bootstrap a democratic election (sole author). (submitted). `https://github.com/mukeshtiwari/Formally_Verified_Verifiable_Group_Generator`.

[3] Modelling Networking Protocols Mathematically. In this work, we develop a formally verified framework that a network protocol designer can use to verify the properties of their protocols (joint work Timothy Griffin. In this work, I formally verified generalised graph algorithm on semiring algebra in Coq theorem prover). (submitted). `https://github.com/mukeshtiwari/Semiring_graph_algorithm`.

[4] Machine Checking the Bayer-Groth Proof of Shuffle. In this work, we formalise the Bayer-Groth Proof of Shuffle, used in many democratic elections to shuffle the ballot (joint work with Thomas Haines, and Rajeev Gore. In this work, I proved facts related to Vandermonde matrix in Coq theorem prover). (major revision USENIX 2023).

+44-7824648138
mt883@cam.ac.uk, mukesh.tiwari@anu.edu.au, mukeshtiwari.iiitm@gmail.com
mukeshtiwari.github.io/ • *mukesh-tiwari-3609486/*
mukeshtiwari

[5] Theorem Provers to Protect Democracies. In this work, we are formalising all the cryptographic components, written in Java, used in Swiss Post in Coq theorem prover. Our goal is to write an independent verifier for the scrutine sheet of elections conducted by Swiss Post software programs (sole author) (work in progress). `https://github.com/mukeshtiwari/Dlog-zkp/`.

[6] Towards Leakage-Resistant Machine Learning in Trusted Execution Environments. In this work, submitted informally in PaveTrust, we develop a formally verified information flow secure (constant-time) gradient descent algorithm with axiomatic differential privacy (sole author) (work in progress). `https://github.com/mukeshtiwari/IFMachine/tree/main/secc/examples/federated_learning`.

[7] Machine Checked Properties of the Schulze Method. In this work, we are formally verifying all the (social choice) properties of the Schulze method. (lead developer) (work in progress). `https://github.com/mukeshtiwari/Schulzeproperties`.

References

- Dirk Pattinson, Research School of Computer Science, Australian National University, Canberra, dirk.pattinson@anu.edu.au

- Thomas Haines, Research School of Computer Science, Australian National University, Canberra, thomas.haines@anu.edu.au

- Toby Murray, School of Computing and Information Systems, University of Melbourne, Melbourne, toby.murray@unimelb.edu.au

- Timothy Griffin, Computer Laboratory, University of Cambridge, Cambridge, tgg22@cam.ac.uk

- Véronique Cortier, Research Director, CNRS, LORIA, Nancy, France, veronique.cortier@loria.fr (I have not worked with her but she understands my work very closely. She has kindly agreed to write me a reference letter based on my election security work)