Van Chan Ngo

School of Computer Science
Carnegie Mellon University, Pittsburgh, PA 15213, USA

Email: channgo@cmu.edu
URL: http://www.cs.cmu.edu/~vanchann/

Current Position

Post-doctoral Researcher, Principles of Programming Languages, School of Computer Science, Carnegie Mellon University, Pittsburgh, USA

Areas of Specialization

Programming Languages • Formal Verification • Compilers • Language-based Security

Research Interests

My research activities aim at building reliable and secure computer systems by developing formal frameworks which guarantee that software satisfies formally its specification, especially embedded safety-critical software such as automotive, avionic, and health-care applications. The construction of a formal framework involves the research and knowledge of principles of programming languages, compiler design and development, and formal methods including model checking, theorem proving, and static analysis, to provide formal assurances that the specification is fulfilled.

Education

08/2014 *Ph.D in Computer Science*, INRIA Rennes-Bretagne Atlantique and Université de Rennes 1, Rennes, France, First Class Honors

06/2008 *MSc in Computer Science*, Université Joseph Fourier, Grenoble, France, under a French Government Scholarship-Évariste Galois

07/2005 *Engineer in Computer Engineering*, Center of Talented Training-PFIEV, Hanoi University of Technology, Hanoi, Vietnam, First Class Honors with Congratulations of the Ministry of Education

Awards & Honors

2017 *Schloss Dagstuhl - NSF Support Grant for Junior Researchers*, Wadern, Saarland, Germany

2011-2014 *Ph.D scholarship*, INRIA France

2007-2008 *DEA scholarship* from the French government, Université de Grenoble, France

2007 *Masters scholarship* from the Italian government, Politecnico di Milano, Italy

2007 *Masters scholarship* from SamSung company, ICU-KAIST, South Korea

2000-2005 *Scholarships* from Hanoi University of Technology for excellent academic results

Employment History

2016-Present *Research Fellow*, School of Computer Science, Carnegie Mellon University, Pittsburgh, USA
• Automatic symbolic resource bound (e.g., time and memory) analysis of functional and imperative programs for detecting security vulnerability including time side-channel attacks, stack overflow, etc
• Static analysis for probabilistic programs, e.g., automatic symbolic expected resource bound analysis such as execution time and memory usage

2015-2016 *Research Engineer*, INRIA, Rennes, France
• Formal verification of probabilistic SystemC models using Statistical Model Checking
• Probabilistic temporal assertion-based verification of SystemC models

2011-2014 *Research Assistant*, INRIA, Rennes, France
• Formal verification of the highly optimizing and industrial synchronous data-flow compiler, Signal, which is used in model-based design of real-time and safety-critical systems

• Using translation validation approach to prove the preservation of clock semantics, data dependencies and value-equivalence for source code and the compiled programs using several techniques including model checking, theorem proving, and graph transformation

2010-2011 *Software Architect*, Tech Propulsion Lab, USA
• Design and development of embedded mobile software on iOS and Android platforms
• Project management in the mobile development department

2008-2009 *Research Assistant*, IBM Zurich Lab and ZISC, Zurich, Switzerland
• Formally verifying and certifying the design of secure boot processes in IBM AIX mainframes
• Formalizing the boot process and its security properties using Promela and the model checker Spin

2007-2008 *Internship*, VERIMAG Laboratory, Grenoble, France
• Automated verification framework for formally proving the IND-CPA security property of asymmetric encryption schemes
• Representing the encryption schemes as *frames* in cryptographic π-calculus, and formalizing the IND-CPA property as a equivalent relation between two frames

2006-2007 *Senior Software Engineer*, IBM, Hanoi, Vietnam
• Working on text search engine of the IBM data base management DB2

2005-2006 *Software Engineer*, FPT Software, Hanoi, Vietnam
• Working on embedded systems for navigation

Technical Strengths

Programming Languages • *Strong knowledge*: C/C++, OCaml • *Familiarity*: Assembly, Java, Python

Toolchains • *Strong knowledge*: GCC, LLVM Compiler Infrastructure

Formal Methods & Modeling Languages • *Strong knowledge*: Synchronous Programming, Logics and Temporal Logics, Model Checking, Theorem Proving, Static Analysis • *Familiarity*: SystemC, Verilog

Formal Method Tools • *Strong knowledge*: SPIN, SMV, UPAAL, PRISM, SMC Plasma Lab, Yices, Z3, Coq, Frama-C

Embedded Software Development • *Familiarity*: AVR, Arduino, RTLinux, FreeRTOS

Languages

• *English*: Advance • *French*: Advance • *Vietnamese*: Native

Publications

According to the core conference ranking, **TOP*** denotes a flagship conference or premium journal (e.g. rank A*) or a premium journal, while **TOP** denotes a good conference or a leading journal (e.g. rank A, B).

Journal & Conference

2018 V.C. Ngo, Q. Carbonneaux, and J. Hoffmann. *Bounded Expectations: Resource Analysis for Probabilistic Programs*. In Proceedings of 2018 ACM SIGPLAN Conference on Programming Language Design and Implementation (**PLDI'18**). ACM, Philadelphia, PA, USA. Acceptance rate: 55/275 = 20.0% **TOP***

2017 V.C. Ngo, M. Dehesa-Azuara, M. Fredrikson, J. Hoffmann. *Verifying and Synthesizing Constant-Resource Implementations with Types*. In Proceedings of 38th IEEE Symposium on Security & Privacy (**SP Oakland'17**). IEEE, San Jose, CA, USA. Acceptance rate: 60/463 = 12.9% **TOP***

2017 V.C. Ngo and A. Legay. *Formal Verification of Probabilistic SystemC Models with Statistical Model Checking*. In Journal of Software: Evolution and Process. Wiley. **TOP**

2016 V.C. Ngo, A. Legay, and V. Joloboff. *PSCV: A Runtime Verification Tool for Probabilistic SystemC Models*. In Proceedings of 28th International Conference on Computer Aided Verification (**CAV'16**). Springer, Toronto, Ontario, Canada. Acceptance rate: 58/195 = 29.7% **TOP***

2016 V.C. Ngo, A. Legay, and J. Quilbeuf. *Statistical Model Checking for SystemC Models*. In Proceedings of 17th High Assurance Systems Engineering Symposium (**HASE'16**). IEEE, Orlando, Florida, USA. Acceptance rate: 32/78 = 41% **TOP**

2015 V.C. Ngo, J-P. Talpin, T. Gautier, L. Besnard, and P. Le Guernic. *Modular Translation Validation of a Full-sized*

Synchronous Compiler using Off-the-shelf Verification Tools. In Proceedings of International Workshop on Software and Compilers for Embedded Systems (**SCOPES'15**). ACM, St. Goar, Germany. Acceptance rate: 8/18 = 44.4% **TOP**

2015 V.C. Ngo, J-P. Talpin, and T. Gautier. *Translation Validation for Synchronous Data-flow Specification in the SIGNAL Compiler*. In Proceedings of 35th IFIP International Conference on Formal Techniques for Distributed Objects, Components and Systems (**FORTE'15**). IFIP, Grenoble, France. Acceptance rate: 15/53 = 28.3% **TOP**

2015 V.C. Ngo, J-P. Talpin, T. Gautier, and P. Le Guernic. *Translation Validation for Clock Transformations in a Synchronous Compiler*. In Proceedings of 18th International Conference on Fundamental Approaches to Software Engineering (**FASE'15**). Springer, London, UK. Acceptance rate: 23/80 = 28.7% **TOP**

2014 V.C. Ngo, J-P. Talpin, and T. Gautier. *Precise Deadlock Detection for Polychronous Data-flow Specifications*. In Proceedings of the Electronic System Level Synthesis Conference (**ESLsyn-DAC'14**). IEEE, San Francisco, CA, USA

2013 V.C. Ngo, J-P. Talpin, T. Gautier, P. Le Guernic, and L. Besnard. *Formal Verification of Synchronous Data-flow Program Transformations Toward Certified Compilers*. In Journal of Frontiers of Computer Science. Special Issue on Synchronous Programming, Springer

2012 V.C. Ngo, J-P. Talpin, T. Gautier, P. Le Guernic, and L. Besnard. *Formal Verification of Automatically Generated C-code from Polychronous Data-flow Equations*. Accepted at International High-Level Design, Validation and Test Workshop (**HLDVT'12**). IEEE, California, USA

2012 V.C. Ngo, J-P. Talpin, T. Gautier, P. Le Guernic, and L. Besnard. *Formal Verification of Compiler Transformations on Polychronous Equations*. In Proceedings of 9th International Conference on Integrated Formal Methods (**IFM'12**). Springer, Pisa, Italy. Acceptance rate: 22/59 = 37% **TOP**

2009 C. Ene, Y. Lakhnech, and V.C. Ngo (Authors by alphabetical order). *Formal Indistinguishability Extended to the Random Oracle Model*. In Proceedings of 14th European Symposium on Research in Computer Security (**ESORICS'09**). Springer, Saint-Malo, France. Acceptance rate: 42/220 = 19.1% **TOP**

2009 C. Ene, Y. Lakhnech, and V.C. Ngo (Authors by alphabetical order). *Formal Indistinguishability Extended to the ROM*. In Proceedings of Workshop on Formal and Computational Cryptography (**FCC'09**), New York, NY, USA

Thesis

2014 V.C. Ngo. *Formal Verification of a Synchronous Data-flow Compiler: from Signal to C*. In Ph.D Thesis in Computer Science, INRIA France, University of Rennes 1, France

2008 V.C. Ngo. *Automated Verification of Asymmetric Encryption*. In MSc Thesis in Computer Science and Applied Mathematics, VERIMAG, University of Grenoble, France

2005 V.C. Ngo. *Theory and Implementation of Distributed Firewall on Linux Environment* (in Vietnamese). In Engineer Thesis in Computer Engineerings, Center for Talent Training, Hanoi University of Technology, Hanoi, Vietnam

Technical Report

2017 V.C. Ngo, Q. Carbonneaux, and J. Hoffmann. *Bounded Expectations: Resource Analysis for Probabilistic Programs*. In CMU, Technical Report

2016 V.C. Ngo, M. Fredrikson, and J. Hoffmann. *Quantifying and Preventing Side Channels with Sub-structural Type Systems*. In CMU, Technical Report

2015 V.C. Ngo, A. Legay, and J. Quilbeuf. *Dependability Analysis of Embedded Control Systems Using SystemC and Statistical Model Checking*. In HAL-INRIA, Technical Report RR-8762

2014 V.C. Ngo, A. Legay, and J. Quilbeuf. *Dynamic Verification of SystemC Specification with Statistical Model Checking*. In HAL-INRIA, Technical Report RR-8644

2014 V.C. Ngo, J-P. Talpin, T. Gautier, and P. Le Guernic. *Evaluating SDVG Translation Validation: from Signal to C*. In HAL-INRIA, Technical Report RR-8508

2012 V.C. Ngo, J-P. Talpin, and P. Le Guernic. *Formal Verification of Transformations on Abstract Clocks in Synchronous Compilers*. In HAL-INRIA, Technical Report RR-8064

2012 V.C. Ngo, J-P. Talpin, T. Gautier, P. Le Guernic, and L. Besnard. *Formal Verification of Synchronous Data-flow Compilers*. In HAL-INRIA, Technical Report RR-7921

Software

Absynth Automatic Bound Synthesizer (Absynth) is a tool that automatically and statically computes upper bounds on the expected resource usage for imperative probabilistic programs

RAML	Resource Aware ML is a tool that automatically and statically computes bounds on resource usage (lower, constant, and upper bounds) for functional programs. It also can check the constant resource-use programs used in preventing timing side-channel attacks
PSCV	A runtime verification tool for probabilistic SystemC models. It consists of two components: the plug-in for Plasma Lab in Java and tool for generating C++ monitor and aspect advices in C++
Plasma Lab	Plasma Lab is a compact, efficient and flexible platform for statistical model checking of stochastic models
Polychrony	The Polychrony tool-set developed in C++ and Java, based on Signal, provides a formal framework to design, develop and validate critical systems, from abstract specification until deployment on distributed systems
SigCert	The tool developed in OCaml checks the correctness of the compilation of Signal compiler w.r.t clock semantics, data dependence, and value-equivalence (not fully implemented)
SigCV	PDS Simulation Relation Checking with SIGALI: implementation of the theory works in IFM 2012 article as the libraries in SIGALI tool-set
Mobile Applications	Mobile applications: RATP, Turnstone, Saigon Places, A86, PhotoEnc,...

Talks

2017	*Bounded Expectations: Resource Analysis for Probabilistic Programs - Tool Demonstration.* Talk at Dagtuhl Seminar 17291, Wadern, Germany, July 2017
2017	*Verifying and Synthesizing Constant-Resource Implementations with Types.* Talk at S&P Oakland 2017, San Jose, CA, USA, May 2017
2016	*PSCV: A Runtime Verification Tool for Probabilistic SystemC Models.* Talk at CAV 2016, Toronto, Ontario, Canada, July 2016
2016	*Statistical Model Checking for SystemC Models.* Talk at HASE 2016, Orlando, FL, USA, January 2016
2015	*Translation Validation for Synchronous Data-flow Specification in the SIGNAL Compiler.* Talk at FORTE-DisCoTec 2015, Grenoble, France, June 2015
2015	*Translation Validation for Clock Transformations in a Synchronous Compiler.* Talk at FASE-ETAPS 2015, London, UK, April 2015
2015	*Dependability Analysis of Embedded Control Systems using SystemC and Statistical Model Checking.* Talk at INRIA Rennes, France, March 2015
2014	*Precise Deadlock Detection for Polychronous Data-flow Specifications.* Talk at ESLsyn-DAC 2014, San Francisco, CA, USA, June 2014
2014	*Seminar: Compilation and Execution of Streaming Programs.* St Germain au Mont d'Or, France, April 2014
2012	*Formal Verification of Transformations on Clocks in Synchronous Data-flow Compilers.* Talk at 19th Open International Workshop on Synchronous Programming 2012, Le Croisic, France, November 2012
2012	*Formal Verification of Transformations on Clocks in Synchronous Data-flow Compilers.* Talk at Beihang University (BUAA), Beijing, China, October 2012
2012	*Formal Verification of Compiler Transformations on Polychronous Equations.* Talk at IFM 2012, Pisa, Italy, June 2012
2009	*Formal Indistinguishability Extended to the Random Oracle Model.* Talk at ESORICS 2009, Le Croisic, France, September 2009

Teaching

2016	*Mechanizing Soundness Proofs of the Automatic Amortized Resource Analysis*, Student project in Computer Science, Carnegie Mellon University
2013	*Introduction to Model Checking*, Teaching assistant, Master in Computer Science, University of Rennes 1
2012	*Automaton-based Modeling and Formal Verification*, Teaching assistant, Master in Computer Science, University of Rennes 1
2000-2003	*High School Student Teaching in Mathematics and Physics*, Tutor, Hanoi University of Technology

Professional Service

2018	*IEEE Embedded Systems Letters* (CEDA), 2018
2018	PC Member: *Developments in Implicit Computational Complexity* (DICE'18), Thessaloniki, Greece, April 2018
2018	*ACM Transactions on Programming Languages and Systems* (TOPLAS), 2018
2017	*The 29th International Conference on Computer Aided Verification* (CAV'17), Heidelberg, Germany, July 2017

2017 *The ACM SIGPLAN Conference on Programming Language Design and Implementation* (PLDI'17), Barcelona, Spain, June 2017

2017 *The 26th International Conference on Compiler Construction* (CC'17), Austin, Texas, US, February 2017

2016 *The 16th International Conference on Formal Methods in Computer-Aided Design* (FMCAD'16), Mountain View, CA, US, October 2016

2016 *The 27th International Conference on Concurrency Theory* (CONCUR'16), Quebec City, Canada, August 2016

2015 *The 13th ACM-IEEE International Conference on Formal Methods and Models for System Design* (MEMOCODE'15), Austin, US, September 2015

2014 *The 8th International Conference on Language and Automata Theory and Applications* (LATA'14), Madrid, Spain, March 2014

Research Projects

STAC This is a DARPA-funded project and will be 48 months in duration. The project involves many Research & Development teams in industry and university research groups, e.g., GrammaTech, Draper, University of Utah, University of Colorado Boulder, Iowa State University, and Carnegie Mellon University. STAC program aims to develop new program analysis techniques and associated tools for identifying vulnerabilities related to the space and time resource usage behavior of algorithms, specifically, vulnerabilities to algorithmic complexity and side-channel attacks. It seeks to enable developers to identify vulnerabilities related to resource usage in software at language levels. The applications will be the information systems that U.S. government, military, and economy depend

DANSE The project focuses on the development of a new methodology to support evolving, adaptive and iterative System of Systems (SoS) life-cycle models based on a formal semantics for SoS inter-operations and supported by novel tools for analysis, simulation, and optimization. DANSE includes industrial representatives with focus on aerospace, land, and automotive systems, as well as a leading tools and framework provider in the system space, and top European research institutes in system engineering. These partners have deep interest in the outcome of the research and are eager to deploy the developments as soon as they become available

DALI The DALI project has undertaken a challenging agenda aimed at extending the people autonomous life beyond the home. The environment where the system operates is partially known (due to its large variability) and changing. Our assisted living device system must therefore acquire dynamic information about the user's immediate environment in order to guide its decision-making. The construction of a system of such complexity represents a major scientific and technological effort bringing together expertise across different disciplines

VERISYNC The project proposed here aims at substantially improving the safety and reliability of embedded software that is being developed in the context of a Model-based design approach. This is achieved by formally proving the correctness of essential transformations that a model undergoes during its compilation to executable code. The definition of the semantics and the correctness proof of the compiler will be carried out by means of theorem proving. The compiler is executable and will be evaluated on realistic examples. The project is targeted at the compilation of a synchronous language to an imperative programming language. Synchronous languages have turned out to be an expressive formalism for embedded algorithms, and their precise semantics make them particularly suitable for our purpose

SCALP Our day-to-day lives increasingly depend upon information and our ability to manipulate it securely. That is, in a way that prevents malicious elements to subvert the available information for their own benefits. This requires solutions based on cryptographic systems (primitives and protocols). However, no matter how carefully crafted cryptographic systems are, experience has shown that effective attacks can remain hidden for years. This may be caused by poor design or often unclear and poorly defined security properties and assumptions. The goal of this project is to achieve a major step towards building automated tools for the verification of cryptographic systems. In order, to reconcile generality, imposed by the high diversity of cryptographic systems, and automation, we shall build our tools upon Coq

AVOTE Electronic voting promises the possibility of a convenient, efficient and secure facility for recording and tallying votes. However, the convenience of electronic elections comes with a risk of large-scale fraud and their security has seriously been questioned. In this project we propose to use formal methods to analyze electronic voting protocols

References

- Jan Hoffmann
  Assistant Professor,
  Computer Science Department, 5000 Forbes Avenue, CMU, PA 15213, USA
  Email: jhoffmann@cmu.edu
  Tel: +1 412 268 6309

- Matthew Fredrikson
  Assistant Professor,
  Computer Science Department & Institute for Software Research, CMU, 5000 Forbes Avenue, PA 15213, USA
  Email: mfredrik@cs.cmu.edu
  Tel: +1 412 268 3992

- Jean-Pierre Talpin
  Research Director,
  TEA: Time, Events and Architectures Group, INRIA Rennes, Campus de Beaulieu, 263 Avenue General Leclerc, 35042 Rennes, France
  Email: Jean-Pierre.Talpin@inria.fr
  Tel: +33 (0)2 99 84 74 36

- Thierry Gautier
  Senior Researcher,
  TEA: Time, Events and Architectures Group, INRIA Rennes, Campus de Beaulieu, 263 Avenue General Leclerc, 35042 Rennes, France
  Email: Thierry.Gautier@inria.fr
  Tel: +33 (0)2 99 84 72 43