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

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

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

2207 *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 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 the language of the model checker Spin, Promela

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

Journal & Conference

2017 V.C. Ngo, Q. Carbonneaux, and J. Hoffmann. *Bounded Expectations: Resource Analysis for Probabilistic Programs.* In Submission to 2018 ACM SIGPLAN Conference on Programming Language Design and Implementation (**PLDI'18**). ACM, Philadelphia, PA, USA [PDF]

2017 V.C. Ngo, M. Dehesa-Azuara, M. Fredrikson, J. Hoffmann. *Verifying and Synthesizing Constant-Resource Implementations with Types.* To appear in 2017 IEEE Symposium on Security & Privacy (**SP Oakland'17**). IEEE, San Jose, CA, USA [PDF]

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 [PDF]

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 [PDF]

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 [PDF]

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 [PDF]

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 [PDF]

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 [PDF]

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 [PDF]

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 [PDF]

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 [PDF]

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 [PDF]

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 [PDF]

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, USA [PDF]

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 [PDF]

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

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

Technical Report

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

2016 V.C. Ngo, M. Fredrikson, and J. Hoffmann. *Quantifying and Preventing Side Channels with Substructural Type Systems.* In CMU, Technical Report [PDF]

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 [PDF]

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 [PDF]

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 [PDF]

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 [PDF]

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 [PDF]

Talks

Academic Conference Presentations: Oakland 2017, CAV 2016, HASE 2016, FORTE 2015, FASE 2015, ESLsyn-DAC 2014, IFM 2012, ESORICS 2009
Invited Presentations: Dagtuhl Seminar 2017, INRIA Rennes 2015, Compilation 2014, Beihang University (BUAA) 2012, Synchron 2012

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 [HTML]
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++ [HTML]
Plasma Lab	Plasma Lab is a compact, efficient and flexible platform for statistical model checking of stochastic models [HTML]
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 [HTML]
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) [HTML]
SigCV	PDS Simulation Relation Checking with SIGALI: implementation of the theory works in IFM 2012 article as the libraries in SIGALI tool-set [HTML]
Mobile Applications	Mobile applications: RATP, Turnstone, Saigon Places, A86, PhotoEnc,... [HTML]

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

Review Activity: DICE 2018, TOPLAS 2018, CAV 2017, PLDI 2017, CC 2017, FMCAD 2016, CONCUR 2016, MEMOCODE 2015, LATA 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 [HTML]
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 [HTML]
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 [HTML]
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 [HTML]

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 [HTML]

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 [HTML]