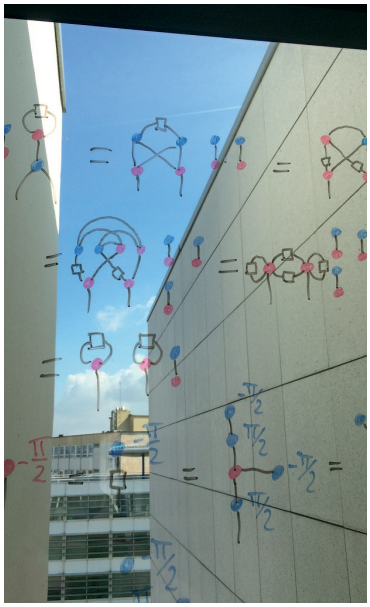
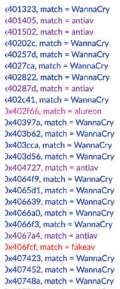
PORTFOLIO

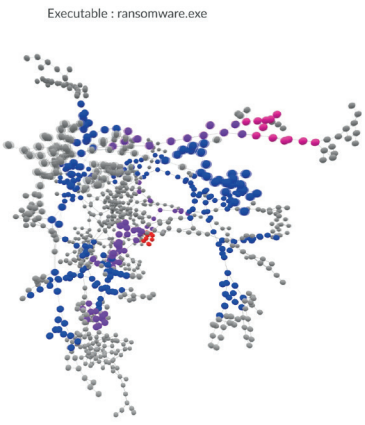
DEPARTMENT 2

Formal methods

BELENIOS







Loria cnrs Inria UNIVERSITÉ DE LORRAINE

En partenariat avec : CentraleSupélec

Portfolio Département 2

Nous présentons dans le portfolio du département 2 “Méthodes formelles” les 7 documents suivants.

Rapport détaillé. L’Université de Lorraine a demandé aux laboratoires de rédiger pour juin 2021 les bilans des laboratoires en vu de l’évaluation HCERES 2016-2020, ceci sans attendre les recommandations de l’HCERES. Nous avons donc rédigé un rapport complet (en anglais), structuré sur la base des anciennes évaluations de l’HCERES. Ce rapport contient une première partie qui présente une vision générale du département et de ses réalisations, suivi d’un rapport détaillé pour chacune des équipes au sein du département. Nous avons naturellement réutilisé pour le DAE une partie substentielle de la première partie, ce qui explique la présence de parties en anglais et en francais dans le DAE. Nous présentons ce rapport complet dans le portfolio.

Nous présentons les six autres éléments suivant du portfolio dans la suite de ce document.

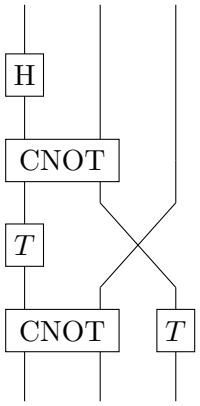
- **Completeness of the ZX-Calculus**. Article publié dans le journal *Logical Methods in Computer Science*.
- **Making explicit domain knowledge in formal system development**. Article publié dans le journal *Science of Computer Programming*.
- **Fifty Shades of Ballot Privacy: Privacy against a Malicious Board**. Article publié à *CSF 2020*.
- **ERC CoG SPOOC: Automated Security Proofs of Cryptographic Protocols**
- **TLAPS: le système de preuve de TLA$^+$**
- **Start-Up Cyber-Detect**

Notons enfin que nous présentons également un zoom sur le **vote electronique et software Belenios**, travail commun aux départements 1 et 2, dans le portfolio du laboratoire.

Complétude du ZX-Calcul

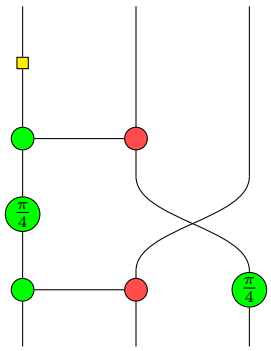
Équipe MOCQUA

En calcul quantique, on représente en général les évolutions sous la forme de *circuits*, dont voici un exemple :



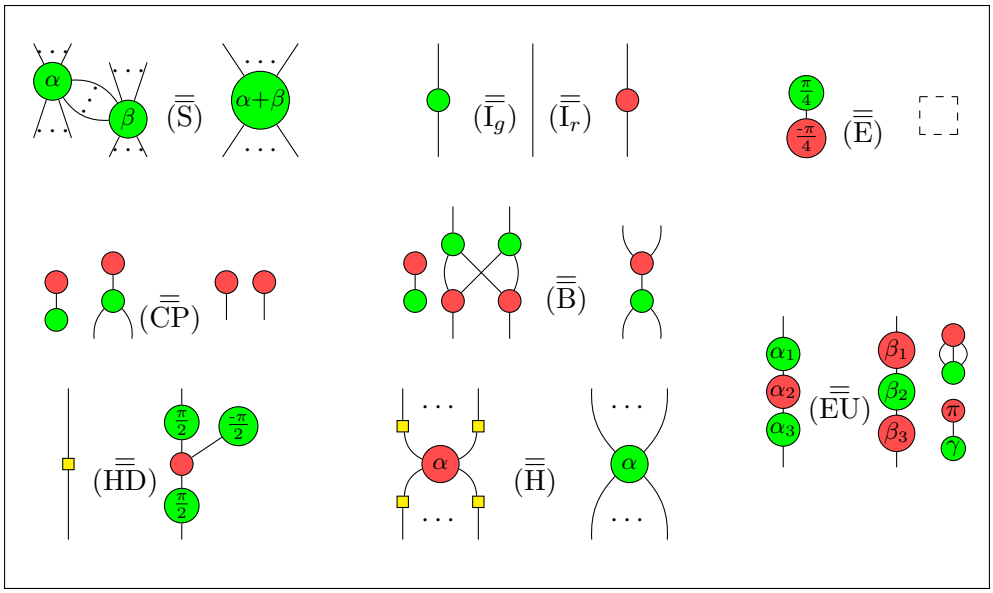
Ce modèle a cependant deux problèmes majeurs : D'abord, un circuit sur n qubits est représenté par une matrice de taille 2^n par 2^n. Donc comprendre son évolution en l'écrivant matriciellement a un surcoût exponentiel. Ensuite, on ne connaît pas à l'heure actuelle l'ensemble des équations qui régissent les circuits, c'est à dire les règles locales qui permettent de passer d'un circuit donné à tout circuit équivalent (représentant la même matrice). En particulier, il y a peu d'espoir de compiler, et d'optimiser des circuits, sans une telle connaissance.

Le ZX-calcul [1] est une alternative aux circuits, qui redécompose certaines portes (comme le CNOT ci dessus) en portes encore plus simples. Ainsi le circuit ci-dessus devient, à une renormalisation près :



On remarque par exemple que la porte CNOT a été transformée en deux portes plus élémentaires, et non unitaires, qu'on peut appliquer en même temps (comme sur le diagramme) ou l'une après l'autre.

Au moment où nous commençons nos travaux, le ZX-calcul a les mêmes problèmes que les circuits : on ne connaît pas d'axiomatisation complète pour le langage en entier, mais uniquement pour des fragments qui ne permettent typiquement pas d'exprimer les circuits importants. C'est ce résultat majeur que nous décrivons ici : la **première axiomatisation complète** du langage. Voici la version la plus simple de l'axiomatisation :



Nous avons obtenu la première axiomatisation complète en 2017, publiée à LICS en 2018 [7]. La version ci-dessus a été proposée par notre doctorant Renaud Vilmart [12]. Elle lui a valu le **prix Kleene** (meilleur papier étudiant) à LICS en 2019, et l'**accessit du prix de thèse Gilles Kahn** en 2020.

La preuve initiale s'inspire de la complétude du ZW-calcul [5], qui ne permet de traiter que des matrices à coefficients entiers. Le résultat initial que nous avons obtenu [7] correspond au fragment Clifford+T du ZX-Calcul, ce qui correspond en terme de circuit aux portes CNOT, H et T vues ci-dessus.

Le résultat de complétude du ZX-calcul a eu un impact très important car il s'agissait d'un problème ouvert depuis plus de 10 ans. En levant ce verrou théorique, la publication de notre résultat a eu un effet important dans la communauté et un groupe de chercheurs à Oxford a immédiatement appliqué notre méthode par encodage pour obtenir une théorie équationnelle complète du ZX-calcul [6] pour toutes les matrices complexes, ce qui nécessite l'introduction de famille infinie d'équations. Nous avons indépendamment obtenue une autre axiomatisation [8], que nous avons ensuite largement simplifié [10, 9], pour arriver à la forme proposée par Renaud ci-dessus.

Plus généralement ce résultat a démontré l'intérêt d'utiliser le ZX-calcul par rapport aux circuits quantiques. Ce résultat de complétude a permis au ZX-calcul de toucher aujourd'hui une communauté plus large, et aux recherches de se focaliser principalement sur les applications du ZX-calcul, par exemple dans l'optimisation de code [11] ou le calcul tolérant aux fautes [3], sujets sur lesquels nous contribuons également [4, 2]. A l'heure actuelle, toutes les meilleures techniques d'optimisation de code quantique sont ainsi basées sur le ZX-calcul.

References

[1] Bob Coecke and Ross Duncan. Interacting quantum observables. In *ICALP*, volume 5126 of *Lecture Notes in Computer Science*, pages 298–310. Springer, 2008.

[2] Niel de Beaudrap, Ross Duncan, Dominic Horsman, and Simon Perdrix. Pauli fusion: a computational model to realise quantum transformations from zx terms. *arXiv preprint arXiv:1904.12817*, 2019.

[3] Niel de Beaudrap and Dominic Horsman. The zx calculus is a language for surface code lattice surgery. *Quantum*, 4:218, 2020.

[4] Ross Duncan, Aleks Kissinger, Simon Perdrix, and John van de Wetering. Graph-theoretic simplification of quantum circuits with the zx-calculus. *Quantum*, 4:279, 2020.

[5] Amar Hadzihasanovic. A diagrammatic axiomatisation for qubit entanglement. In *2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 573–584, Kyoto, July 2015. ACM/IEEE.

[6] Amar Hadzihasanovic, Kang Feng Ng, and Quanlong Wang. Two complete axiomatisations of pure-state qubit quantum computing. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 502–511. ACM, 2018.

[7] Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. A complete axiomatisation of the ZX-calculus for Clifford+T quantum mechanics. In *ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 559–568, 2018.

[8] Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. Diagrammatic reasoning beyond Clifford+T quantum mechanics. In *ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 569–578, 2018.

[9] Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. A generic normal form for ZX-diagrams and application to the rational angle completeness. In *ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2019.

[10] Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. Completeness of the ZX-calculus. *Logical Methods in Computer Science*, 16(2):11:1–72, 2020.

[11] Aleks Kissinger and John van de Wetering. Reducing the number of non-clifford gates in quantum circuits. *Physical Review A*, 102(2):022406, 2020.

[12] Renaud Vilmart. A Near-Minimal Axiomatisation of ZX-Calculus for Pure Qubit Quantum Mechanics. In *LICS 2019 - 34th Annual ACM/IEEE Symposium on Logic in Computer Science*, Vancouver, Canada, June 2019.

Explicitation du domaine d'expertise en développement formel de systèmes

Équipe MOSEL-VERIDIS

Les systèmes logiciels évoluent et s'exécutent dans un environnement ou contexte. Raisonner sur la correction de leur comportement repose sur une relation ternaire entre les modèles de besoins, de systèmes et de contexte. Les méthodes formelles offrent des outils (automatiques) pour la synthèse et l'analyse de tels modèles et se sont intéressées à des relations binaires : validation d'un modèle vis-à-vis d'un modèle informel, vérification d'un modèle formel vis-à-vis d'un modèle formel, génération de code à partir d'un modèle, génération de tests à partir de besoins, etc. Le contexte, dans ces cas, est traité comme de seconde classe : en général il est *implicite* et réparti entre besoins et modèles. Notre travail [1] montre comment rendre *explicite* la modélisation des contextes et des environnements associés à des domaines d'application.

En général, *explicite* signifie *clairement exprimé* (visiblement) alors que *implicite* signifie *indirectement exprimé* ou impliqué. Notons que dans le génie logiciel, la signification de ces termes fait apparaître des inconsistances. En analyse des besoins, ces termes sont utilisés pour distinguer l'expression déclarative (descriptive) et opérationnelle (prescriptive) des besoins. Cette phase d'analyse consiste à générer des besoins explicites (de niveau type) et déclaratifs à partir de besoins implicites (de niveau instance ou scénario). Le besoin de méthodes formelles pour une telle génération est clair et les travaux consistent à traiter formellement les termes implicites et explicites dans le processus d'ingénierie du logiciel ou des systèmes.

Plusieurs approches [2, 3, 4, 5] traitent de la formalisation de théories mathématiques utilisées pour le développement formel de systèmes. Ces théories contribuent à la construction de formalisations complexes exprimant et réutilisant des preuves de propriétés. En général, ces théories sont définies au sein de contextes qui sont importés ou instanciés. Elles offrent un cadre pour représenter la sémantique implicite du système à développer et sont fondées sur la logique, algèbres, théorie des types, etc.

Ces travaux traitent, de manière intégrée, la description formelle de domaines constituant des contextes dans lesquels les systèmes évoluent. Par exemple, les propriétés dépendant du contexte (poids dépendant de la gravité) ne sont pas exprimées dans la théorie où les développements sont menés. Ce type d'information est exprimé dans une sémantique explicite. Plusieurs propriétés importantes sont vérifiées en méthode formelle. Elles sont définies et vérifiées à partir de la sémantique implicite associée à la technique formelle utilisée : contrôle de types, preuve, réécriture, raffinement, model checking, analyse de traces, simulation etc. Lorsque ces propriétés sont traitées dans leur contexte en leur associant une sémantique explicite, elles peuvent ne plus être valides. Un exemple est la composition de systèmes échangeant des flottants représentant des monnaies exprimées en dollars dans l'un et en euros dans l'autre. L'absence de sémantique explicite dans le contexte de preuve rend cette composition invalide. Ainsi, les activités de développement doivent être reconsidérées en fonction de la possibilité de représenter non seulement la sémantique implicite mais aussi la sémantique explicite. La formalisation de ces opérations de développement en séparant l'implicite de l'explicite renforcerait la correction des systèmes ainsi développés. Cet aspect constitue un problème significatif si l'on souhaite développer des systèmes dynamiques, à base de composants hétérogènes fiables, dans des contextes qui ne le sont pas. Ainsi, nous traitons la séparation des aspects intrinsèques et extrinsèques en permettant la construction des modèles formels de contextes par l'utilisation des méthodes formelles fondées sur la preuve au travers de plusieurs domaines

d'application comme le domaine médical [6], le domaine aéronautique [7], le domaine du vote [8] ou encore des IHM [9]. Le diagramme de la figure suivante résume les différentes tâches et leur intégration:

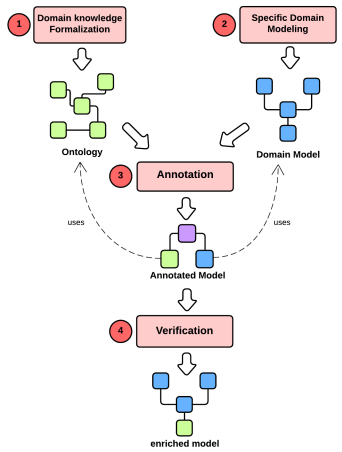


Schéma général de la démarche

Ces travaux ont été largement diffusés au travers de projets ANR comme IMPEX, Formedicis et DISCONT mais aussi un colloque international a réuni des spécialistes de méthodologie formelle au centre de Shonan en 2016 [10] avec un livre rassemblant les propositions [11]. Dans le cadre du projet Formedicis, les travaux [12, 9, 13, 14] ont été menés pour définir une approche méthodologique liée au développement des IHM conforme à des standards de ce domaine. Dans le cadre du domaine médical, nous avons aussi utilisé les connaissances pour redéfinir les modèles que nous avions développé auparavant [6]. Pour conclure, nous avons mis en évidence trois mécanismes de structuration des modèles orientés à états. En premier lieu, le mécanisme d'annotation définit la relation entre les modèles ontologiques et les modèles orientés à états et permet ainsi d'importer des connaissances ontologiques dans les modèles orientés à états. En second lieu, les ontologies sont utilisées pour refactoriser les modèles orientés à états. En troisième lieu, le mécanisme de dépendance de modèles permet d'organiser le développement des modèles orientés à états et de définir les constantes des modèles par construction. Enfin, un certain nombre d'études de cas comme le pacemaker et le vote illustrent la méthodologie développée. Un plugin de formalisation d'ontologies dans des contextes Event-B a été développé et permet de mettre en œuvre le mécanisme d'annotation.

References

[1] Yamine Aït Ameur and Dominique Méry. Making explicit domain knowledge in formal system development. *Sci. Comput. Program.*, 121:100–127, 2016.

[2] Michael Jackson and Pamela Zave. Domain descriptions. In *Proceedings of IEEE International Symposium on Requirements Engineering, RE 1993, San Diego, California, USA, January 4-6, 1993*, pages 56–64. IEEE, 1993.

[3] Dines Bjørner. Domain analysis and description principles, techniques, and modelling languages. *ACM Trans. Softw. Eng. Methodol.*, 28(2):8:1–8:67, 2019.

[4] Dines Bjørner. *Domain Science and Engineering - A Foundation for Software Development*. Monographs in Theoretical Computer Science. An EATCS Series. Springer, 2021.

[5] Dines Bjørner. Domain analysis & description - the implicit and explicit semantics problem. In Régine Laleau, Dominique Méry, Shin Nakajima, and Elena Troubitsyna, editors, *Proceedings Joint Workshop on Handling IMPlicit and EXplicit knowledge in formal system development (IMPEX) and Formal and Model-Driven Techniques for Developing Trustworthy Systems (FM&MDD), IMPEX/FM&MDD 2017, Xi'an, China, 16th November 2017*, volume 271 of *EPTCS*, pages 1–23, 2017.

[6] Neeraj Kumar Singh, Yamine Aït-Ameur, and Dominique Méry. Formal Ontological Analysis for Medical Protocols. In *Implicit and explicit semantics integration in proof based developments of discrete systems*. Springer, January 2021.

[7] Dominique Méry, Rushikesh Sawant, and Anton Tarasyuk. Integrating domain-based features into event-b: A nose gear velocity case study. In Ladjel Bellatreche and Yannis Manolopoulos, editors, *Model and Data Engineering - 5th International Conference, MEDI 2015, Rhodes, Greece, September 26-28, 2015, Proceedings*, volume 9344 of *Lecture Notes in Computer Science*, pages 89–102. Springer, 2015.

[8] Dominique Méry and Souad Kherroubi. Contextual Dependency in State-based Modelling. In *Implicit and explicit semantics integration in proof based developments of discrete systems*. Springer, January 2021.

[9] Neeraj Kumar Singh, Yamine Aït-Ameur, Ismail Mendil, Dominique Méry, David Navarre, Philippe Palanque, and Marc Pantel. F3FLUID: A formal framework for developing safety-critical interactive systems in FLUID. *Journal of Software: Evolution and Process*, 2022. Early View = Online Version of Record before inclusion in an issue : e2439.

[10] Yamine Aït Ameur, Shin Nakajima, and Dominique Méry. Implicit and explicit semantics integration in proof based developments of discrete systems (NII shonan meeting 2016-16). *NII Shonan Meet. Rep.*, 2016, 2016.

[11] Yamine Aït-Ameur, Shin Nakajima, and Dominique Méry. *Implicit and Explicit Semantics Integration in Proof-Based Developments of Discrete Systems*. Springer Singapore, 2021.

[12] Neeraj Kumar Singh, Yamine Aït-Ameur, Romain Geniet, Dominique Méry, and Philippe Palanque. On the Benefits of Using MVC Pattern for Structuring Event-B Models of WIMP Interactive Applications. *Interacting with Computers*, May 2021.

[13] Ismail Mendil, Yamine Aït-Ameur, Neeraj Kumar Singh, Dominique Méry, and Philippe Palanque. Standard Conformance-by-Construction with Event-B. In Alberto Lluch Lafuente and Anastasia Mavridou, editors, *FMICS 2021 - 26th International Conference on Formal Methods for Industrial Critical Systems*, volume 12863 of *Formal Methods for Industrial Critical Systems. 26th International Conference, FMICS 2021, Paris, France, August 24–26, 2021, Proceedings ; ISBN 978-3-030-85247-4*, pages 126–146, Paris, France, August 2021. Springer International Publishing.

[14] Ismail Mendil, Yamine Aït-Ameur, Neeraj Kumar Singh, Dominique Méry, and Philippe Palanque. Leveraging Event-B Theories for Handling Domain Knowledge in Design Models. In Shengchao Qin, Jim Woodcock, and Wenhui Zhang, editors, *SETTA 2021 - 7th International Symposium on Dependable Software Engineering. Theories, Tools, and Applications*, volume 13071 of *Lecture Notes in Computer Science*, pages 40–58, Beijing/Online, China, November 2021. Springer International Publishing.

Fifty shades of ballot privacy

Équipe PESTO

Le vote électronique

Le vote électronique essaie de garantir les mêmes propriétés de sécurité que le vote papier à l'urne, tel qu'il est organisé par exemple en France lors des grandes élections. *A minima*, on souhaite le secret du vote (nul ne doit savoir comment j'ai voté) et la transparence du scrutin, souvent appelée vérifiabilité : un électeur doit pouvoir vérifier que son vote a bien été compté et que seuls des bulletins légitimes ont été acceptés et comptabilisés.

Comme pour les protocoles de sécurité en général, les bonnes pratiques requièrent qu'un protocole soit *prouvé sûr* avant d'être déployé. Ainsi, la Chancellerie Suisse [2] formule l'exigence suivante pour la mise en place d'un vote électronique : "une preuve symbolique et une preuve cryptographique doivent attester que le protocole cryptographique respecte les exigences visées." Or un pré-requis pour prouver qu'un protocole respecte les propriétés souhaitées est de *définir* formellement sa sécurité. Si les propriétés classiques d'authentification ou de secret d'une clé sont bien comprises, les propriétés de vérifiabilité et de secret du vote donnent du fil à retordre à la communauté scientifique. Nous avions ainsi proposé une synthèse des différents modèles existants pour la vérifiabilité, en discutant de leurs subtilités [3]. En général, la vérifiabilité s'appuie sur une *urne publique*, consultable par tous les électeurs, qui peuvent s'assurer que leur bulletin de vote (chiffré) est bien présent. Des techniques cryptographiques (les preuves zero-knowledge) permettent de s'assurer que le résultat correspond aux bulletins chiffrés, sans connaître les clés de déchiffrement.

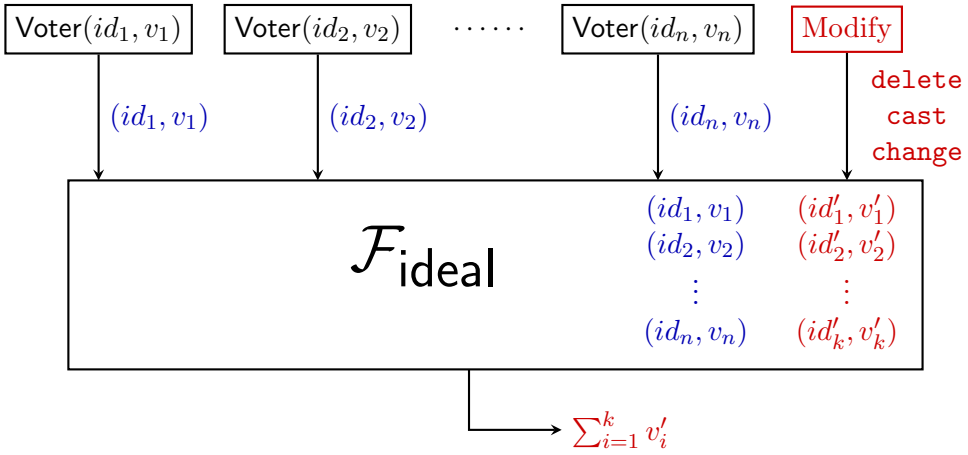
Les difficultés du secret du vote

Curieusement, définir le secret du vote s'avère encore plus complexe. Une raison est que le résultat d'une élection révèle forcément un peu d'information sur le vote d'un électeur. Ainsi, si le résultat proclame qu'un candidat a été élu à l'unanimité, alors le vote de chaque électeur est révélé. On peut bien sûr s'accommoder de cette situation mais il faut alors définir le secret du vote, en tolérant les cas où on considère que la perte du secret est inévitable. Nous avons ainsi montré [1] que de nombreuses définitions du secret du vote pouvaient être mises en défaut, soit parce qu'elles pouvaient déclarer sûrs des protocoles attaqués, soit parce qu'elles ne permettaient pas de couvrir de nombreux protocoles, soit encore parce qu'elles n'étaient pas réalisables. Nous avons alors proposé une nouvelle définition, appelé BPRIV.

Différentes nuances d'urne

Malheureusement, toutes ces définitions, y compris BPRIV, supposent implicitement que le serveur de l'élection, ou du moins l'urne publique, est honnête au sens où les bulletins des électeurs ne sont aucunement manipulés. C'est une situation bien regrettable puisque les protocoles de vote sont justement conçus pour assurer le secret du vote en présence d'un serveur malhonnête !

Définir le secret du vote en présence d'une urne malhonnête est un défi car une urne malhonnête pourrait ne garder que le bulletin de l'électeur dont on souhaite connaître le vote. Nous avons ainsi montré dans des travaux précédents [4] qu'il est impossible d'obtenir le secret du vote sans un minimum de vérifiabilité : le système doit permettre d'assurer que les votes des électeurs sont bien comptés. Pour prendre en compte une urne malhonnête, il faut accepter certains comportements, par exemple le fait qu'un attaquant puisse retirer des votes et donc apprendre plus d'information sur un sous-ensemble de votes. Nous avons proposé de nous comparer à un système idéal, sans cryptographie, où l'on décrit précisément ce qu'on tolère : la suppression de bulletins, le réordonnancement des bulletins, la manipulation des bulletins, etc. Ce système idéal est représenté ci-après.



On dit alors qu'un protocole de vote est sûr si un attaquant n'apprend pas plus d'information que sur le système idéal, pour les manipulations autorisées. Suivant ce qu'on tolère, on obtient différentes nuances du secret du vote. Prouver qu'un protocole de vote est aussi sûr que le système idéal est assez fastidieux. Pour faciliter la tâche, nous avons proposé une définition sous forme de jeu, mBPRIV, dans l'esprit de BPRIV, qui implique le secret du vote basé sur un système idéal. Nous avons ainsi pu étudier plusieurs protocoles de la littérature et caractériser la notion de secret du vote qu'ils garantissent, comme résumé ci-dessous. Ce travail a été récompensé par un **distinguished paper award** à la conférence CSF 2020.

	$\mathcal{F}^{\emptyset}_{\text{ideal}}$		$\mathcal{F}^{\text{del}}_{\text{ideal}}$	$\mathcal{F}^{\text{del,reorder}}_{\text{ideal}}$	$\mathcal{F}^{\text{del,reorder,change}}_{\text{ideal}}$
	General case	all voters check			
Helios - no revote	✗	✓	✗	✗	✓
Helios - revote	✗	✗	✗	✗	✗
Belenios - no revote	✗	✓	✓	✓	✓
Belenios - revote	✗	✓	✓	✓	✓
Civitas - revote	✗	✓	✓	✓	✓

Perspectives

Une autre approche pour éviter de considérer le cas d'une urne malhonnête est ... de s'assurer que l'urne est honnête. Pour cela, l'approche la plus classique est de distribuer l'urne sur plusieurs

serveurs, dont on pourra supposer qu'un certain nombre se comporte comme attendu. Nous avons caractérisé [5] les propriétés que doit vérifier une urne pour être suffisamment "sûre" dans le cadre du vote électronique. Lors de ce travail, nous avons montré que plusieurs systèmes actuels ne réalisent pas ces propriétés et peuvent être attaqués.

Définir les bonnes propriétés d'un système de vote électronique est un sujet de recherche très actif dans l'équipe. Plus généralement, nous nous attachons à modéliser, prouver et concevoir des systèmes de vote, dont notre plateforme Belenios. Nous évoquons ces sujets dans la fiche "vote électronique : impact socio-économique" du laboratoire.

References

[1] David Bernhard, Véronique Cortier, David Galindo, Olivier Pereira, and Bogdan Warinschi. A comprehensive analysis of game-based ballot privacy definitions. In *36th IEEE Symposium on Security and Privacy (S&P'15)*, San Jose, United States, May 2015.

[2] Swiss Confederation. Technical and administrative requirements for electronic vote casting. Annex to the FCh Ordinance of 13 December 2013 on Electronic Voting (OEV, SR 161.116), 2018.

[3] Véronique Cortier, David Galindo, Ralf Kuesters, Johannes Mueller, and Tomasz Truderung. SoK: Verifiability Notions for E-Voting Protocols. In *36th IEEE Symposium on Security and Privacy (S&P'16)*, San Jose, United States, May 2016.

[4] Véronique Cortier and Joseph Lallemand. Voting: You Can't Have Privacy without Individual Verifiability. Research report, CNRS, Inria, LORIA, August 2018.

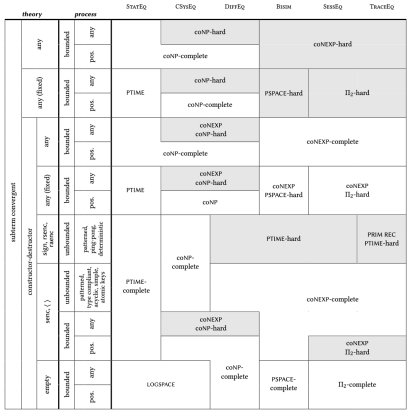
[5] Lucca Hirschi, Lara Schmid, and David Basin. Fixing the Achilles Heel of E-Voting: The Bulletin Board. In *CSF 2021 - 34th IEEE Computer Security Foundations Symposium*, pages 1–17, Dubrovnik/Virtual, Croatia, June 2021. IEEE.

ERC CoG SPOOC
Automated Security Proofs of Cryptographic Protocols:
Privacy, Untrusted Platforms and Applications to E-voting Protocols

Équipe PESTO

The security of electronic transactions is ensured by cryptographic protocols. While historically their main goals were confidentiality and authentication the situation has changed. The ability of people to stay connected constantly combined with ill-conceived systems seriously threatens people's privacy. Due to viruses and malware, personal computers and mobile phones must not be considered trustworthy anymore; yet they execute the protocols that are to achieve security goals. The goals of the SPOOC project were to develop solid foundations and practical tools to analyse and formally prove security protocols that ensure user privacy as well as techniques for executing protocols on untrusted platforms. For this the project was structured around three themes:

- foundations and practical tools for specifying and verifying new security properties, in particular privacy properties;
- techniques for design and automated analysis of protocols that can be executed on untrusted platforms;
- application of these methods in particular to novel e-voting protocols, that aim for strong security guarantees without need to trust the voter client software.



Complexity of process equivalences

Foundations for verifying indistinguishability properties
Many security properties are expressed as equivalences in cryptographic process calculi. However, nearly all existing work focused on (much simpler) reachability properties. We significantly advanced both the theoretical understanding and practical verification. We obtained an extensive picture of the complexity for the verification of equivalences [8, 11] showing notably that verification of trace and observational equivalence is coNEXP-complete for a large class of cryptographic primitives when bounding the number of sessions. Despite the high complexity we were able to develope several complementary (in terms of expressivity, precision and efficiency) algorithms and tools for automated verification.

- AKISS [5] is based on a novel Horn clause resolution algorithm. It supports many cryptographic primitives (including XOR [2]), else branches [20], but bounded replication. The tool is precise for a class of determinate processes, and can approximate equivalence otherwise. It suffers from inefficiency when the number of sessions increases. The paper describing AKISS [5] is listed in *ACM Computing Reviews' 21st Annual Best of Computing list of notable books and articles for 2016*.
- SAT-Equiv [12] implements a novel algorithm, based on graph planning and SAT-solving. It supports only standard cryptographic primitives, requires protocols to be determinate and

replication to be bounded and does not support else branches. The tool is however extremely efficient, allowing to verify a very large number of sessions.

- DEEPSEC [8, 9, 10] is based on a novel constraint solving algorithm. It supports many cryptographic primitives, and the whole applied pi calculus, only bounding replication. It is precise and has good efficiency (slightly less than SAT-Equiv) for the class of determinate processes (where partial order reductions apply). It also supports equivalence by session: a very efficient proof technique that allows for partial-order and symmetry reductions. [8] received a *distinguished paper award for at S&P'18*.

- TYPE-EQ [15, 16] is based on a novel type system for proving equivalence properties. It supports only standard cryptographic primitives and no private channels, but allows analysis of an unbounded number of sessions. The tool is not complete, i.e. it may provide false attacks. It is however extremely efficient.

Mechanisms to achieve security guarantees on untrusted platforms and global state When machines are compromised, e.g. infected by a malware, two main approaches allow to nevertheless achieve some level of security: put the human in the loop, or rely on trusted hardware. These two approaches come with new challenges for formal verification.

In modern security protocols humans may be required to compare or copy short strings between devices. We propose a symbolic model which takes into account that short strings may be subject to brute-force attacks and propose a new decision procedure integrated in the AKISS tool and tested on protocols from the ISO/IEC 9798-6:2010 standard [18].

Hardware security modules require to maintain a *global state* which was only poorly supported by existing tools. We proposed SAPiC, Stateful Applied Pi Calculus, for specifying such protocols, and an eponymous verification tool [21]. We also designed in SAPiC a framework for reasoning about Isolated Execution Environments (IEEs), such as Intel SGX, which offer the possibility to execute sensitive code in isolation. The SAPiC tool [21] has been integrated as a plugin into Tamarin, a state-of-the-art protocol analysis tool. We also enhanced SAPiC with support for liveness properties [1]. In the underlying Tamarin prover we improved support for equational theories [19] and automation [13]. The improvement of automation [13] received a *best paper award at ESORICS'20*. The Tamarin prover was also used to analyze the (stateful) AKA authentication protocol of the 5G standard and discovered that some critical security goals were not met [3]. In [7], we developed a front-end to the popular ProVerif tool to take into account global state, which was widely believed to be an intrinsic limitation of this tool: as a result nearly all protocols formally verified in the literature and relying on global state can be handled by this tool, more efficiently and with better automation.

Application to e-voting protocols When analyzing e-voting protocols we were confronted to the fact that existing security definitions were not entirely satisfying. We have designed new definitions for what it means for an e-voting protocol to be secure, even if the underlying platform or the authorities are untrusted [4, 17]. The definitions in [17] represent the most comprehensive framework for the analysis of electronic voting schemes, and received a *distinguished paper award at CSF'20*.

We have analysed Du-Vote, a recent malware tolerant remote electronic voting scheme. It uses a hardware token to provide high security guarantees, both vote privacy and verifiability. We

provide an extensive security analysis of Du-Vote and show several attacks on both privacy as well as verifiability [22]. We have also formally verified the e-voting protocol deployed in the Swiss canton of Neuchâtel [14] for which we analysis mostly confirm the security.

Finally, we designed and implemented a new verifiable voting scheme, BeleniosRF [6], that is receipt-free in a strong sense: even dishonest voters cannot prove how they voted.

Highlights

- 4 distinguished papers in major conferences and a journal (CSF, ESORICS, S&P, ToCL).
- We laid the foundations for the verification of equivalence properties and built practical tools, in particular the DEEPSEC prover.
- Contributions to the state-of-the-art tools Tamarin (which is now developed at ETH Zurich, CISPA and PESTO) and ProVerif.
- Verification of real-life deployed protocols: the Neuchâtel e-voting and 5G AKA protocols.
- 4 PhD students and 3 post-docs were part of SPOOC.

References

[1] Michael Backes, Jannik Dreier, Steve Kremer, and Robert Künnemann. A Novel Approach for Reasoning about Liveness in Cryptographic Protocols and its Application to Fair Exchange. In *2nd IEEE European Symposium on Security and Privacy (EuroS&P'17)*, Proceedings of the 2nd IEEE European Symposium on Security and Privacy, Paris, France, April 2017. Springer.

[2] David Baelde, Stéphanie Delaune, Ivan Gazeau, and Steve Kremer. Symbolic verification of privacy-type properties for security protocols with XOR. In *CSF 2017 - 30th IEEE Computer Security Foundations Symposium*, page 15, Santa Barbara, United States, August 2017. IEEE.

[3] David Basin, Jannik Dreier, Lucca Hirschi, Saša Radomirovic, Ralf Sasse, and Vincent Stettler. A Formal Analysis of 5G Authentication. In *ACM CCS 2018 - 25th ACM Conference on Computer and Communications Security*, volume 14 of *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, Toronto, Canada, October 2018. ACM Press.

[4] Sergiu Bursuc, Constantin-Catalin Dragan, and Steve Kremer. Private votes on untrusted platforms: models, attacks and provable scheme. In *EuroS&P 2019 - 4th IEEE European Symposium on Security and Privacy*, Stockholm, Sweden, June 2019.

[5] Rohit Chadha, Vincent Cheval, Ştefan Ciobâcà Ciobâcă, and Steve Kremer. Automated verification of equivalence properties of cryptographic protocols. *ACM Transactions on Computational Logic*, 17(4), 2016.

[6] Pyrros Chaidos, Véronique Cortier, Georg Fuchsbauer, and David Galindo. BeleniosRF: A Non-interactive Receipt-Free Electronic Voting Scheme. In *23rd ACM Conference on Computer and Communications Security (CCS'16)*, Vienna, Austria, October 2016.

[7] Vincent Cheval, Véronique Cortier, and Mathieu Turuani. A little more conversation, a little less action, a lot more satisfaction: Global states in ProVerif. Research report, Inria Nancy -

Grand Est ; LORIA, UMR 7503, Université de Lorraine, CNRS, Vandoeuvre-lès-Nancy, April 2018.

[8] Vincent Cheval, Steve Kremer, and Itsaka Rakotonirina. DEEPSEC: Deciding Equivalence Properties in Security Protocols - Theory and Practice. In *39th IEEE Symposium on Security and Privacy*, San Francisco, United States, May 2018.

[9] Vincent Cheval, Steve Kremer, and Itsaka Rakotonirina. The DEEPSEC prover. In *CAV 2018 - 30th International Conference on Computer Aided Verification*, Oxford, United Kingdom, July 2018.

[10] Vincent Cheval, Steve Kremer, and Itsaka Rakotonirina. Exploiting Symmetries When Proving Equivalence Properties for Security Protocols. In *CCS'19 - 26th ACM Conference on Computer and Communications Security*, London, United Kingdom, November 2019.

[11] Vincent Cheval, Steve Kremer, and Itsaka Rakotonirina. The hitchhiker's guide to decidability and complexity of equivalence properties in security protocols. In Nigam, V., Ban Kirigin, T., Talcott, C., Guttman, J., Kuznetsov, S., Thau Loo, B., Okada, and M., editors, *Logic, Language, and Security. Essays Dedicated to Andre Scedrov on the Occasion of His 65th Birthday*, volume 12300 of *Lecture Notes in Computer Science*, Philadelphia, United States, 2020. Springer.

[12] Véronique Cortier, Antoine Dallon, and Stéphanie Delaune. SAT-Equiv: An Efficient Tool for Equivalence Properties. In *30th IEEE Computer Security Foundations Symposium (CSF'17)*, pages 481 – 494, Santa Barbara, United States, July 2017. IEEE.

[13] Véronique Cortier, Stéphanie Delaune, and Jannik Dreier. Automatic generation of sources lemmas in Tamarin: towards automatic proofs of security protocols. In *ESORICS 2020 - 25th European Symposium on Research in Computer Security*, volume 12309 of *Lecture Notes in Computer Science*, pages 3–22, Guilford, United Kingdom, September 2020.

[14] Véronique Cortier, David Galindo, and Mathieu Turuani. A formal analysis of the Neuchâtel e-voting protocol. In *EuroS&P 2018 - 3rd IEEE European Symposium on Security and Privacy*, Londres, United Kingdom, April 2018.

[15] Véronique Cortier, Niklas Grimm, Joseph Lallemand, and Matteo Maffei. A Type System for Privacy Properties. In *CCS'17 - 24th ACM Conference on Computer and Communications Security*, pages 409 – 423, Dallas, United States, October 2017.

[16] Véronique Cortier, Niklas Grimm, Joseph Lallemand, and Matteo Maffei. Equivalence Properties by Typing in Cryptographic Branching Protocols. In *POST'18 - 7th International Conference on Principles of Security and Trust*, Thessaloniki, Greece, April 2018.

[17] Véronique Cortier, Joseph Lallemand, and Bogdan Warinschi. Fifty Shades of Ballot Privacy: Privacy against a Malicious Board. In *CSF 2020 - 33rd IEEE Computer Security Foundations Symposium*, Boston / Virtual, United States, June 2020.

[18] Stéphanie Delaune, Steve Kremer, and Ludovic Robin. Formal verification of protocols based on short authenticated strings (extended version). Research report, Inria Nancy - Grand Est, 2017.

[19] Jannik Dreier, Charles Duménil, Steve Kremer, and Ralf Sasse. Beyond Subterm-Convergent Equational Theories in Automated Verification of Stateful Protocols (extended version). In *POST 2017 - 6th International Conference on Principles of Security and Trust*, volume 10204 of *Proceedings of the 6th International Conference on Principles of Security and Trust*, pages 117–140, Uppsala, Sweden, April 2017. Springer.

[20] Ivan Gazeau and Steve Kremer. Automated analysis of equivalence properties for security protocols using else branches. In *22nd European Symposium on Research in Computer Security (ESORICS'17)*, Oslo, Norway, 2017. Springer.

[21] Steve Kremer and Robert Künnemann. Automated Analysis of Security Protocols with Global State. *Journal of Computer Security*, 24(5), 2016.

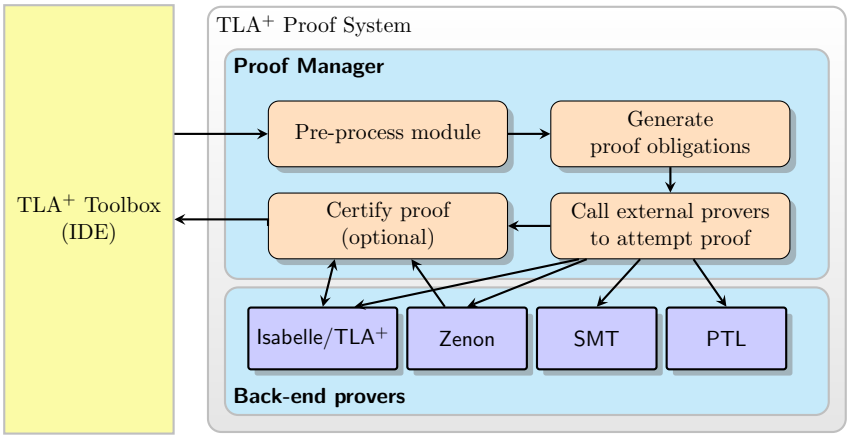
[22] Steve Kremer and Peter Rønne. To Du or not to Du: A Security Analysis of Du-Vote. In *IEEE European Symposium on Security and Privacy 2016*, Proceedings of the IEEE European Symposium on Security and Privacy 2016, Saarbrucken, Germany, March 2016. IEEE Computer Society.

The TLA$^+$ Proof System

Équipe MOSEL-VERIDIS

TLA$^+$ [8] is a specification language originally designed for specifying concurrent and distributed systems and their properties. The language is based on TLA, a linear-time temporal logic, and on Zermelo-Fraenkel set theory with the axiom of choice. Writing and verifying TLA$^+$ specifications is supported by the TLA$^+$ Toolbox, an integrated development environment based on Eclipse that facilitates the use of the tools associated with TLA$^+$, including the TLC model checker [14] and TLAPS, the TLA$^+$ proof system [5]. TLA$^+$ has been used successfully in academia as well as in industry (e.g., [10, 12]).

With support from the Joint Inria-Microsoft Research Centre, the Mosel-VeriDis team has played a leading role in the design and development of TLAPS, a proof assistant for carrying out proofs about TLA$^+$ specifications. TLA$^+$ proofs are written in a hierarchical proof language, and the language and TLAPS itself have been designed to be independent of particular theorem provers: all interaction takes place at the level of TLA$^+$, letting the user focus on the specification of the algorithm being developed. The following figure illustrates the overall architecture of the system.



Architecture of TLAPS.

The proof manager interprets a TLA$^+$ module containing theorems and proofs. In particular, it tracks the logical context (declared symbols and usable facts) and generates proof obligations for every step of a proof. It interfaces with external back-end theorem provers and translates proof obligations into their input language. The main proof back-ends are (i) interfaces to SMT solvers through an encoding into the SMT-lib language [3], (ii) the tableau-based Zenon prover [4] with custom rules for set theory of TLA$^+$, (iii) an encoding of TLA$^+$ into the logical framework Isabelle [11], and (iv) a decision procedure for propositional temporal logic [13]. If one of the back-end provers discharges the proof obligation, the proof is accepted, but it can optionally be certified using the most trusted backend Isabelle/TLA$^+$ (this functionality is currently implemented only for Zenon).

The inclusion of powerful SMT solvers as backends for TLAPS significantly improved the degree of automation that users experience. The main challenge in designing that backend was to devise a sound and efficient translation from the set theory of TLA$^+$, which is untyped and contains second-order constructs, into the multi-sorted first-order logic underlying SMT-lib [9]. The separation of

non-temporal proof steps (which represent more than 95% of a typical TLA$^+$ proof) and temporal reasoning is achieved by a technique called *coalescing* [6], an on-the-fly abstraction of subformulas contained in proof obligations for either first-order or temporal logic backends. Finally, an important challenge for proving liveness properties is to handle ENABLED predicates that are building blocks of fairness hypotheses and characterize states in which certain transitions may be taken. The basic idea is to replace the ENABLED operator of TLA$^+$ by existential quantification over the variables representing the successor state, but for better automation it is important to simplify the resulting formulas, and we designed a rewriting system that helps establish quantifier-free formulas in many practical cases, contributing to highly automatic proofs. Besides improvements of the coverage of the TLA$^+$ language and a more comprehensive library of lemmas about operators defined in standard modules, future work on TLAPS will aim at better automation based on stronger type inference, extennded capabilities for certifying proofs, the selective expansion of defined operators, and an integration with symbolic model checking techniques.

TLAPS has been presented at several tutorials and events colocated with international conferences. It has been used for carrying out several significant verification projects, including the proof of determinacy of a high-level model of a real-time operating system kernel [1], the verification of a variant of the Pastry distributed hash table protocol [2], an analysis of electronic voting protocols involving a colleague from the Pesto team [7], and the verification of a reconfiguration protocol for the distributed data base system CosmosDB [12].

Just like the other tools in the TLA$^+$ ecosystem, TLAPS is distributed under a permissive open-source license.[1] A Google group lets users exchange questions and helpful tips, and a yearly workshop provides a forum for the larger TLA$^+$ community.

References

[1] Selma Azaiez, Damien Doligez, Matthieu Lemerre, Tomer Libal, and Stephan Merz. Proving determinacy of the pharos real-time operating system. In Michael J. Butler, Klaus-Dieter Schewe, Atif Mashkoor, and Miklós Biró, editors, *5th Intl. Conf. Abstract State Machines, Alloy, B, TLA, VDM, and Z (ABZ 2016)*, volume 9675 of *Lecture Notes in Computer Science*, pages 70–85, Linz, Austria, 2016. Springer.

[2] Noran Azmy, Stephan Merz, and Christoph Weidenbach. A machine-checked correctness proof for pastry. *Sci. Comput. Program.*, 158:64–80, 2018.

[3] Clark Barrett, Pascal Fontaine, and Cesare Tinelli. The SMT-LIB standard: Version 2.6. https://smtlib.cs.uiowa.edu/language.shtml, 2021.

[4] Richard Bonichon, David Delahaye, and Damien Doligez. Zenon: An extensible automated theorem prover producing checkable proofs. In Nachum Dershowitz and Andrei Voronkov, editors, *14th Intl. Conf. Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, volume 4790 of *LNCS*, pages 151–165, Yerevan, Armenia, 2007. Springer.

[5] Denis Cousineau, Damien Doligez, Leslie Lamport, Stephan Merz, Daniel Ricketts, and Hernán Vanzetto. TLA$^+$ proofs. In Dimitra Giannakopoulou and Dominique Méry, editors, *18th Intl. Symp. Formal Methods (FM 2012)*, volume 7436 of *LNCS*, pages 147–154, Paris, France, 2012. Springer.

[1] TLAPS is available at https://tla.msr-inria.inria.fr/tlaps/content/Home.html.

[6] Damien Doligez, Jael Kriener, Leslie Lamport, Tomer Libal, and Stephan Merz. Coalescing: Syntactic abstraction for reasoning in first-order modal logics. In Christoph Benzmüller and Jens Otten, editors, *Automated Reasoning in Quantified Non-Classical Logics*, number CoRR abs/1409.3819 in The Computing Research Repository, Vienna, Austria, September 2014.

[7] Lucca Hirschi, Lara Schmid, and David A. Basin. Fixing the Achilles heel of e-voting: The bulletin board. In *34th IEEE Computer Security Foundations Symposium (CSF 2021)*, pages 1–17, Dubrovnik, Croatia, 2021. IEEE.

[8] Leslie Lamport. *Specifying Systems*. Addison-Wesley, Boston, Mass., 2002.

[9] Stephan Merz and Hernán Vanzetto. Encoding TLA$^+$ into unsorted and many-sorted first-order logic. *Sci. Comput. Program.*, 158:3–20, 2018.

[10] Chris Newcombe, Tim Rath, Fan Zhang, Bogdan Munteanu, Marc Brooker, and Michael Deardeuff. How Amazon Web Services uses formal methods. *CACM*, 58(4):66–73, 2015.

[11] Lawrence C. Paulson. *Isabelle: A Generic Theorem Prover*, volume 828 of *Lecture Notes in Computer Science*. Springer Verlag, 1994.

[12] William Schultz, Ian Dardik, and Stavros Tripakis. Formal verification of a distributed dynamic reconfiguration protocol. In Andrei Popescu and Steve Zdancewic, editors, *11th ACM SIGPLAN Intl. Conf. Certified Programs and Proofs (CPP 2022)*, pages 143–152, Philadelphia, PA, USA, 2022. ACM.

[13] Martin Suda. Variable and clause elimination for LTL satisfiability checking. *Math. Comput. Sci.*, 9(3):327–344, 2015.

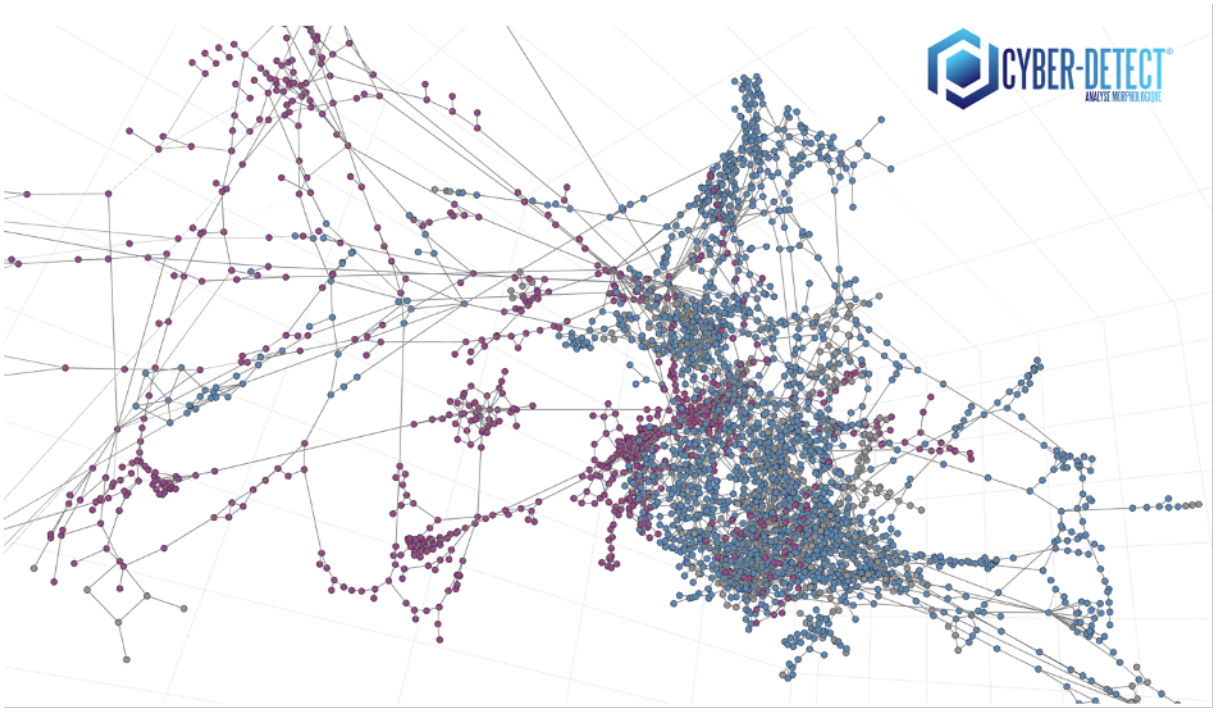
[14] Yuan Yu, Panagiotis Manolios, and Leslie Lamport. Model checking TLA+ specifications. In L. Pierre and T. Kropf, editors, *Correct Hardware Design and Verification Methods (CHARME'99)*, volume 1703 of *Lecture Notes in Computer Science*, pages 54–66, Bad Herrenalb, Germany, 1999. Springer Verlag.

Création de la start-up Cyber-Detect dans le domaine de l'analyse des malwares

Équipe CARBONE

Cyber-Detect est une spin-off du Loria depuis mai 2017. Cyber-Detect développe et commercialise des solutions logicielles pour la détection et l'analyse de menaces types malware. Ces solutions proviennent des travaux de recherches de l'équipe Carbone du département 2.

Innovation. La principale contribution des chercheurs du Loria est l'invention de l'analyse morphologique qui consiste à définir le graphe de contrôle de flot d'un programme comme une signature l'identifiant. Cette idée de départ a été raffinée essentiellement vers deux directions. La première direction construit une abstraction du contrôle de graphe qui préserve la signature. Ensuite, la seconde direction est de ne considérer que des parties du graphe de contrôle de flot. Chaque partie formant une micro-signature d'une fonctionnalité particulière, appelée **site**. Dès lors, la reconnaissance d'un comportement malveillant se fait en détectant des sites. La mise au point du premier prototype a nécessité l'exploitation du LHS pour tester et valider la solution sur les bases de données de malwares.



Transfert. La suite logicielle d'analyse morphologique a été déposée à l'APP et la SATT Sayens a transféré les droits d'exploitation à la start-up Cyber-Detect. Un projet Rapid DGA (2018-2022) a été obtenu pour poursuivre le développement entre Cyber-Detect et l'équipe Carbone du Loria. Cyber-Detect participe également au projet européen Concordia.

Exploitation Aujourd'hui Cyber-Detect a une dizaine d'employés et commercialise trois produits:

- Analyse forensique post-mortem : **GORILLE Expert** destiné aux experts en rétro-ingénierie;
- Prévention et détection : **GORILLE Cloud**;
- Détection d'attaque dormante : **GORILLE Patrol**, une suite logicielle permettant de déporter le scan anti-virus d'une machine distante sur un serveur.

Suite Les membres de Carbone ont poursuivi leurs travaux dans différentes directions : (i) en travaillant sur la conception d'outils d'analyse statique pour améliorer la déobfuscation, (ii) les approches de *machine learning* sont venues compléter les heuristiques de détection et (iii) en renfonçant les outils d'analyse dynamique. Les travaux sont en général publiés dans des conférences de rang A comme Usenix, Security & Privacy, CCS, Saner. La suite des travaux, en collaboration avec l'équipe Resist du département 3, a amené à créer en 2021 un Labcom CNRS avec Wallix, et à répondre à plusieurs appels d'offres de la BPI. Enfin, l'Université de Lorraine pilote le projet *programmes malveillants* du PEPR cybersécurité.