Activity Report 2018

Project-Team INDES

Secure Diffuse Programming

RESEARCH CENTER
Sophia Antipolis - Méditerranée

THEME
Distributed programming and Software engineering

Table of contents

Project-Team INDES

Creation of the Team: 2009 January 01, updated into Project-Team: 2010 July 01

Keywords:

Computer Science and Digital Science:

A1.3. - Distributed Systems
A2. - Software
A2.1. - Programming Languages
A2.1.1. - Semantics of programming languages
A2.1.3. - Object-oriented programming
A2.1.4. - Functional programming
A2.1.7. - Distributed programming
A2.1.9. - Synchronous languages
A2.1.12. - Dynamic languages
A2.2.1. - Static analysis
A2.2.5. - Run-time systems
A2.2.9. - Security by compilation
A4. - Security and privacy
A4.3.3. - Cryptographic protocols
A4.6. - Authentication
A4.7. - Access control
A4.8. - Privacy-enhancing technologies

Other Research Topics and Application Domains:

B6.3.1. - Web
B6.4. - Internet of things
B9.5.1. - Computer science
B9.10. - Privacy

1. Team, Visitors, External Collaborators

Research Scientists

Manuel Serrano [Team leader, Inria, Senior Researcher, HDR]
Nataliia Bielova [Inria, Researcher]
Ilaria Castellani [Inria, Researcher]
Tamara Rezk [Inria, Researcher, HDR]

Post-Doctoral Fellows

Yoon Seok Ko [Inria, from Oct 2018]
Nguyen Nhat Minh Ngo [Inria]
Doliere Some [Inria, from Nov 2018]

PhD Students

Imane Fouad [Inria]
Jayanth Krishnamurthy [Inria, from Sep 2018]
Heloise Maurel [Inria, from Oct 2018]
Bertrand Petit [Inria]

Doliere Some [Inria, until Oct 2018]
Colin Vidal [Inria, until Jun 2018]

Interns

Thibaud Ardoin [Inria, from Jun 2018 until Jul 2018]
Sadry Fievet [Inria, from Apr 2018 until Sep 2018]

Administrative Assistant

Nathalie Bellesso [Inria]

Visiting Scientists

Mauricio Cano [University of Groningen, from Feb 19, 2018 until Apr 21, 2018]
Paola Giannini [University of Piemonte Orientale, from Feb 19, 2018 until Mar 3, 2018]

External Collaborators

Marc Feeley [University of Montréal, Aug 2018]
Gérard Berry [Collège de France, HDR]

2. Overall Objectives

2.1. Overall Objectives

The goal of the Indes team is to study models for diffuse computing and develop languages for secure diffuse applications. Diffuse applications, of which Web 2.0 applications are a notable example, are the new applications emerging from the convergence of broad network accessibility, rich personal digital environment, and vast sources of information. Strong security guarantees are required for these applications, which intrinsically rely on sharing private information over networks of mutually distrustful nodes connected by unreliable media.

Diffuse computing requires an original combination of nearly all previous computing paradigms, ranging from classical sequential computing to parallel and concurrent computing in both their synchronous / reactive and asynchronous variants. It also benefits from the recent advances in mobile computing, since devices involved in diffuse applications are often mobile or portable.

The Indes team contributes to the whole chain of research on models and languages for diffuse computing, going from the study of foundational models and formal semantics to the design and implementation of new languages to be put to work on concrete applications. Emphasis is placed on correct-by-construction mechanisms to guarantee correct, efficient and secure implementation of high-level programs. The research is partly inspired by and built around Hop, the web programming model proposed by the former Mimosa team, which takes the web as its execution platform and targets interactive and multimedia applications.

3. Research Program

3.1. Parallelism, concurrency, and distribution

Concurrency management is at the heart of diffuse programming. Since the execution platforms are highly heterogeneous, many different concurrency principles and models may be involved. Asynchronous concurrency is the basis of shared-memory process handling within multiprocessor or multicore computers, of direct or fifo-based message passing in distributed networks, and of fifo- or interrupt-based event handling in web-based human-machine interaction or sensor handling. Synchronous or quasi-synchronous concurrency is the basis of signal processing, of real-time control, and of safety-critical information acquisition and display. Interfacing existing devices based on these different concurrency principles within HOP or other diffuse programming languages will require better understanding of the underlying concurrency models and of the way they can nicely cooperate, a currently ill-resolved problem.

3.2. Web and functional programming

We are studying new paradigms for programming Web applications that rely on multi-tier functional programming. We have created a Web programming environment named HOP. It relies on a single formalism for programming the server-side and the client-side of the applications as well as for configuring the execution engine.

HOP is a functional language based on the SCHEME programming language. That is, it is a strict functional language, fully polymorphic, supporting side effects, and dynamically type-checked. HOP is implemented as an extension of the BIGLOO compiler that we develop. In the past, we have extensively studied static analyses (type systems and inference, abstract interpretations, as well as classical compiler optimizations) to improve the efficiency of compilation in both space and time.

3.3. Security of diffuse programs

The main goal of our security research is to provide scalable and rigorous language-based techniques that can be integrated into multi-tier compilers to enforce the security of diffuse programs. Research on language-based security has been carried on before in former Inria teams. In particular previous research has focused on controlling information flow to ensure confidentiality.

Typical language-based solutions to these problems are founded on static analysis, logics, provable cryptography, and compilers that generate correct code by construction. Relying on the multi-tier programming language HOP that tames the complexity of writing and analysing secure diffuse applications, we are studying language-based solutions to prominent web security problems such as code injection and cross-site scripting, to name a few.

4. New Software and Platforms

4.1. Bigloo

KEYWORD: Compilers
FUNCTIONAL DESCRIPTION: Bigloo is a Scheme implementation devoted to one goal: enabling Scheme based programming style where C(++) is usually required. Bigloo attempts to make Scheme practical by offering features usually presented by traditional programming languages but not offered by Scheme and functional programming. Bigloo compiles Scheme modules. It delivers small and fast stand alone binary executables. Bigloo enables full connections between Scheme and C programs, between Scheme and Java programs.

RELEASE FUNCTIONAL DESCRIPTION: modification of the object system (language design and implementation), new APIs (alsa, flac, mpg123, avahi, csv parsing), new library functions (UDP support), new regular expressions support, new garbage collector (Boehm's collection 7.3alpha1).

- Participant: Manuel Serrano
- Contact: Manuel Serrano
- URL: http://www-sop.inria.fr/teams/indes/fp/Bigloo/

4.2. Hop

KEYWORDS: Programming language - Multimedia - Iot - Web 2.0 - Functional programming
SCIENTIFIC DESCRIPTION: The Hop programming environment consists in a web broker that intuitively combines in a single architecture a web server and a web proxy. The broker embeds a Hop interpreter for executing server-side code and a Hop client-side compiler for generating the code that will get executed by the client.

An important effort is devoted to providing Hop with a realistic and efficient implementation. The Hop implementation is validated against web applications that are used on a daily-basis. In particular, we have developed Hop applications for authoring and projecting slides, editing calendars, reading RSS streams, or managing blogs.

FUNCTIONAL DESCRIPTION: Multitier web programming language and runtime environment.

- Participant: Manuel Serrano
- Contact: Manuel Serrano
- URL: http://hop.inria.fr

4.3. IFJS

Infomation Flow monitor inlining for JavaScript

KEYWORD: Cybersecurity

FUNCTIONAL DESCRIPTION: The IFJS compiler is applied to JavaScript code. The compiler generates JavaScript code instrumented with checks to secure code. The compiler takes into account special features of JavaScript such as implicit type coercions and programs that actively try to bypass the inlined enforcement mechanisms. The compiler guarantees that third-party programs cannot (1) access the compiler internal state by randomizing the names of the resources through which it is accessed and (2) change the behaviour of native functions that are used by the enforcement mechanisms inlined in the compiled code.

- Contact: Tamara Rezk
- URL: http://www-sop.inria.fr/indes/ifJS/

4.4. iflowsigs.js

KEYWORDS: Compilers - Monitoring

FUNCTIONAL DESCRIPTION: iflowsigs.js is a JavaScript library designed to inline an information flow monitor into JavaScript code. iflowsigs.js support is able to track information flow even in programs that interact with arbitrary Web APIs.

- Participants: José Fragoso Santos and Tamara Rezk
- Contact: Tamara Rezk
- URL: http://j3fsantos.github.io/PersonalPage/IFMonitor/

4.5. iflowTYPES.js

FUNCTIONAL DESCRIPTION: iflowtypes.js is a JavaScript library designed to type secure information flow in JavaScript. iflowtypes.js has two main modes of operation: fully static and hybrid. In the hybrid mode, the program to be typed is instrumented with runtime assertions that are verified at runtime. By deferring rejection to runtime, the hybrid type system is able to type more programs than fully static mechanisms.

- Contact: Tamara Rezk
- URL: http://j3fsantos.github.io/PersonalPage/TypeSystem/

4.6. Mashic

KEYWORD: Security

FUNCTIONAL DESCRIPTION: The Mashic compiler is applied to mashups with untrusted scripts. The compiler generates mashups with sandboxed scripts, secured by the same origin policy of the browsers. The compiler is written in Bigloo.

- Contact: Tamara Rezk
- URL: http://web.ist.utl.pt/~ana.matos/Mashic/mashic.html

4.7. Hiphop.js

KEYWORDS: Web 2.0 - Synchronous Language - Programming language
FUNCTIONAL DESCRIPTION: HipHop.js is an Hop.js DLS for orchestrating web applications. HipHop.js helps programming and maintaining Web applications where the orchestration of asynchronous tasks is complex.

- Contact: Manuel Serrano
- URL: http://hop-dev.inria.fr/hiphop

4.8. Server-Side Protection against Third Party Web Tracking

KEYWORDS: Privacy - Web Application - Web - Architecture - Security by design - Program rewriting techniques
FUNCTIONAL DESCRIPTION: We present a new web application architecture that allows web developers to gain control over certain types of third party content. In the traditional web application architecture, a web application developer has no control over third party content. This allows the exchange of tracking information between the browser and the third party content provider.

To prevent this, our solution is based on the automatic rewriting of the web application in such a way that the third party requests are redirected to a trusted third party server, called the Middle Party Server. It may be either controlled by a trusted party, or by a main site owner and automatically eliminates third-party tracking cookies and other technologies that may be exchanged by the browser and third party server

- Contact: Doliere Some
- URL: http://www-sop.inria.fr/members/Doliere.Some/essos/

4.9. BELL

Browser fingerprinting via Extensions and Login-Leaks
KEYWORDS: Browser Extensions - Security and Privacy in Web Services - Social Networks Security and Privacy
FUNCTIONAL DESCRIPTION: Recent studies show that users can be tracked based on their web browser properties. This software is designed to conduct an experiment on such kinds of user tracking. In this experiment, we demonstrate that a Web user can also be tracked by

- her browser extensions (such as AdBlock, Pinterest, or Ghostery), and

- the websites she has logged in (such as Facebook, Gmail, or Twitter).

In the experiment, we collect user's browser fingerprint, together with the browser extensions installed and a list of websites she has logged in. We only collect anonymous data during the experiment (more details in our Privacy Policy [1]), we will securely store the data on an Inria server, use it only for research purposes and not share it with anyone outside of Inria.

- Contact: Gabor Gulyas
- URL: https://extensions.inrialpes.fr/

4.10. webstats

Webstats
KEYWORDS: Web Usage Mining - Statistic analysis - Security

[1] https://extensions.inrialpes.fr/privacy.php

FUNCTIONAL DESCRIPTION: The goal of this tool is to perform a large-scale monthly crawl of the top Alexa sites, collecting both inline scripts (written by web developers) and remote scripts, and establishing the popularity of remote scripts (such as Google Analytics and jQuery). With this data, we establish whether the collected scripts are actually written in a subset of JavaScript by analyzing the different constructs used in those scripts. Finally, we collect and analyze the HTTP headers of the different sites visited, and provide statistics about the usage of HTTPOnly and Secure cookies, and the Content Security Policy in top sites.

- Contact: Doliere Some
- URL: https://webstats.inria.fr

4.11. Platforms

4.11.1. Skini

Skini is a distributed system for composing and producing live performances with audience participation using HTML5 devices. Skini is developed in Hop and HipHop. It proposes a framework for composing music based on patterns and dynamic control of the orchestration for either synthesizers or musicians.

The system has two basic concepts, "Pattern" and "Orchestration" which can be modulated as much as the composer wants in terms of complexity, duration, etc. The platform is meant for interaction with the audience during the show. Each participant can interact with the platform as well as with other participants. According to our experience during the Golem show in MANCA 2017, we implemented five levels of interaction, which allowed theoretically 120 possible combinations.

5. New Results

5.1. Information Flow Security

We have pursued our study on information flow security policies and enforcements. We have followed two main axes.

Impossibility of Precise and Sound Termination Sensitive Security Enforcements An information flow policy is termination sensitive if it imposes that the termination behavior of programs is not influenced by confidential input. Termination sensitivity can be statically or dynamically enforced. On one hand, existing static enforcement mechanisms for termination sensitive policies are typically quite conservative and impose strong constraints on programs like absence of while loops whose guard depends on confidential information. On the other hand, dynamic mechanisms can enforce termination sensitive policies in a less conservative way. Secure Multi-Execution (SME) , one of such mechanisms, was even claimed to be sound and precise in the sense that the enforcement mechanism will not modify the observable behavior of programs that comply with the termination sensitive policy. However, termination sensitivity is a subtle policy, that has been formalized in different ways. A key aspect is whether the policy talks about actual termination, or observable termination.

We have proved that termination sensitive policies that talk about actual termination are not enforceable in a sound and precise way. For static enforcements, the result follows directly from a reduction of the decidability of the problem to the halting problem. However, for dynamic mechanisms the insight is more involved and requires a diagonalization argument.

In particular, our result contradicts the claim made about SME. We correct these claims by showing that SME enforces a subtly different policy that we call indirect termination sensitive noninterference and that talks about observable termination instead of actual termination. We construct a variant of SME that is sound and precise for indirect termination sensitive noninterference. Finally, we also show that static methods can be adapted to enforce indirect termination sensitive information flow policies (but obviously not precisely) by constructing a sound type system for an indirect termination sensitive policy.

This study is described in [16].

A Better Facet of Dynamic Information Flow Control

Multiple Facets (MF) is a dynamic enforcement mechanism which has proved to be a good fit for implementing information flow security for JavaScript. It relies on multi executing the program, once per each security level or view, to achieve soundness. By looking inside programs, MF encodes the views to reduce the number of needed multi-executions.

In this year, we have published a paper [15], where we have extended Multiple Facets in three directions. First, we propose a new version of MF for arbitrary lattices, called Generalised Multiple Facets, or GMF. GMF strictly generalizes MF, which was originally proposed for a specific lattice of principals. Second, we propose a new optimization on top of GMF that further reduces the number of executions. Third, we strengthen the security guarantees provided by Multiple Facets by proposing a termination sensitive version that eliminates covert channels due to termination.

5.2. JavaScript Implementation

We have pursued the development of Hop.js and our study on efficient JavaScript implementation. We have followed three main axes.

Implementing Hop.js

Hop.js supports full ECMAScript 5 but it still lack many of the new features ECMAScript 2016 has introduced and now that are now well established in ECMAScript 2017. During the year, we have implemented many of these features (iterators, destructuring assignments, modules, etc.). Few constructs remain missing and hopefully will be added to the system by the end of the year (map, set, and proxies). Completing full ECMAScript 2017 is important as we now see more and more packages using these new features being made available and we consider that maintaining the ability to use all these publicly available resources is a prerequisite to a wide Hop.js adoption. We also consider that this is an important asset for Hop.js users, in particular, for the Denimbo company, an Inria startup using Hop.js extensively.

Ahead-of-time JavaScript compilation

Hop.js differs from most JavaScript implementations by many aspects because contrary to all fast and popular JavaScript engines that use just-in-time compilation, Hop.js relies on static compilation, *a.k.a.,* ahead-of-time (AOT) compilation. It is an alternative approach that can combine good speed and lightweight memory footprint, and that can accommodate read-only memory constraints that are imposed by some devices and some operating systems. Unfortunately the highly dynamic nature of JavaScript makes it hard to compile statically and all existing AOT compilers have either gave up on good performance or full language support.

Indeed, JavaScript is hard to compile, much harder than languages like C, Java, and even harder than Scheme and ML two other close functional languages. This is because a JavaScript source code accepts many more possible interpretations than other languages do. It forces JavaScript compilers to adopt a defensive position by generating target codes that can cope with all the possible, even unlikely, interpretations because general compilers can assume very little about JavaScript programs. The situation is worsened further by the *raise as little errors as possible* principle that drives the design of the language. JavaScript functions are not required to be called with the declared number of arguments, fetching an unbound property is permitted, assigning undeclared variables is possible, etc.

All these difficulties are considered serious enough to prevent classic static compilers to deliver efficient code for a language as dynamic and as flexible as JavaScript. We do not share this point of view. We think that by carefully combining classical analyses, by developing new ones when needed, and by crafting a compiler where the results of the high-level analyses are propagated up to the code generation, it is possible for AOT compilation to be in the same range of performance as fast JIT compilers. This is what we attempt to demonstrate with this study. Of course, our ambition is not to produce a compiler strictly as fast as the fastest industrial JavaScript implementations. This would require much more engineering strength than what we can afford. Instead, we only aim at showing that static compilation can have performances reasonably close to those of fastest JavaScript implementations. *Reasonably close* is of course a subjective notion, that everyone is free to set for himself. For us, it means a compiler showing half the performances of the fastest implementations.

The version of the Hop.js AOT compiler we have developed during the year contains new typing analyses and heuristics that compensate for the lack of information JavaScript source codes contain. A first analysis, named *occurrence typing*, that elaborates on top of older techniques developed for the compilation of the Scheme programming language, extracts as much as possible syntactic information directly out of the source code. This analysis alone would give only rough approximations of the types used by the program but its main purpose is to feed the compiler with sufficient information so that it can deploys more efficient supplemental analyses. Probably the most original one is the analysis that we have named *hint typing* or *which typing* that consists in assigning types to variables and to function arguments according to the efficiency of the generated code. In other words, the *which typing* assign types for which the compiler will be able to deliver its best code instead of assigning types that might denote all the possible values variables and arguments may have during all possible executions. We have shown that these *whiched* types correspond very frequently to the implicit *intentional* types programmers had in mind when they wrote their programs. These analyses and the optimizations they enable are implemented in Hop.js version 3.2.0 available on the Inria pages and from Github. They are described in [17] paper.

Property caches: Property caches are a well-known technique invented over 30 years ago to improve dynamic object accesses. They have been adapted to JavaScript, which they have greatly contributed to accelerate. However, this technique is applicable only when some constraints are satisfied by the objects, the properties, and the property access sites. We have started a study to try to improve it on two common usage patterns: *prototype accesses* and *megamorphic accesses*. We have built a prototypical implementation in Hop.js that has let us measure the impact of the technique we propose. We have observed that they effectively complement traditional caches and that they reduce cache misses and consequently accelerate execution. Moreover, they do not cause a slowdown in the handling of the other usage patterns. We are now at completing this study by polishing the implementation and by publishing a paper exposing and evaluating the new techniques.

5.3. Web Reactive Programming

During the year, we have continued our effort in designing and implementing the HipHop.js programming language, we have applied it to interactive music composition, and we have studied security of reactive systems.

HipHop.js

Web applications react to many sort of events. Let them be GUI events, multimedia events, or network events on client code or IO and system events on the server, they are all triggered asynchronously. JavaScript, the hegemonic programming language of the Web, handles them using low level constructs based on *listeners*, a synonym for *callback*. To improve on the so-called *callback hell*, the recent versions of the language have proposed new constructs that raise the programming abstraction level (promises and `async/await`). They enable a programming style, closer to traditional sequential programming, which helps developing and maintaining applications. However, the improvements they propose rely exclusively on syntactic extensions. They do not change the programming model. For that reason, complex orchestration problems that imply all sorts of synchronization, preemption, and parallelism remain as complex to program as before. We think that orchestration should be reconsidered more globally and from the ground. The solution we propose consists in embedding a DSL specialized on orchestration inside the traditional Web development environment, in our case, Hop.js, the Web programming language that the team develop.

The orchestration DSL we propose is called HipHop.js. It is a reactive synchronous language. More precisely, it is an adaptation of the Esterel programming language to the Web. The motivations for choosing Esterel are diverse. First, and most important, Esterel is powerful enough to handle all the orchestration patterns we are considering. Second, the team, via its partnership with Colège de France, has high expertise in the design and development of Esterel-like languages, which constitutes a highly valuable asset for our development.

Esterel is powerful enough to handle all the orchestration patterns we are considering but Esterel has been designed and developed in a context baring no resemblance with the Web. Esterel was considering static execution models while the Web assumes permanent evolutions and modifications of the running programs. Esterel was considering sequential imperative languages for its embedding, while the Web is considering dynamic functional languages (i.e., JavaScript). Esterel was assuming static execution contexts where *a-priori* validity proof were enforced before hand while the Web assumes highly dynamic runtime executions so that only dynamic verifications are doable. For all these reasons, adapting Esterel and transforming it to form HipHop.js has needed a deep revamping and a deep paradigm shift.

During the year of 2018, we have finalized and completed the design of the language that is now almost stabilized. It follows previous version developed in C. Vidal's PhD studies [13], [18]. The version 0.3.x has been made available at the URL http://hop-dev.inria.fr. It has been used to implement our first orchestration demanding applications, in particular, an interactive music composition application. Our next steps will consist in completing the design and implementation of the language and a minimal development environment without which only experts can use the system. We of course also need to publicize the system and describe its design and internal in various academic publications.

Interactive music composition: the Skini platform

In the sixties, the philosopher Umberto Eco, and musicians such as K. Stockausen, K. Penderescki, L. Berio questioned about the relationship between composers, musicians, and the way we perceive music. Eco used the wording "Open Work", and showed that, the vision of the world evolved from a static world to a more blurred perception. According to this new perception and in a shift comparable to the evolution of physics from Copernic to Einstein, some contemporary artists tried to express this complexity through works where the performer and the audience have a concrete impact on the work. Since the sixties, the development of audience participation for collaborative music production has become a more and more active field. Thanks to the large device market and web based technology development such as web audio API, "Open Work" got a broader meaning with systems allowing individual interaction. Nevertheless it is still difficult to find systems proposing frameworks dedicated to music composition of interactive performances with a clear composition scheme and ease of use. This is our motivation for developing a framework, called Skini, designed for composing, simulating, and executing interactive performances. Skini is based on elementary music patterns, automatic control of the patterns activation made possible thanks to Hop and Hiphop. Skini was first used for a concert that took place at the very end of 2017 in the contemporary Musical Festival of Nice (MANCA) followed in 2018 by performances during the "Portes ouvertes" of Inria, the "Fête de la Science" and the Synchron conference.

In 2018: The Skini's user interface has been revamped. We have tried several interfaces for the pattern activation and focused on a simple one in order to make the interface more intuitive and fluid. We have added an important feature called the "distributed sequencer", which allows the audience not only to activate patterns but also to create them. We have added a new level of interaction, the scrutator, which allows global actions by the audience on the orchestration. The complete system is now synchronized with an external Midi clock. We have developed a first version of stand alone Midi control of the pattern. The system has followed the evolution of the Hiphop syntax and now implements the last version for the control of orchestration. We improved the synchronisation system and the processes for implementing the orchestration.

5.4. Session Types

Session types describe communication protocols between two or more participants by specifying the sequence of exchanged messages, together with their functionality (sender, receiver and type of carried data). They may be viewed as the analogue, for concurrency and distribution, of data types for sequential computation. Originally conceived as a static analysis technique for an enhanced version of the π-calculus, session types have now been embedded into a range of functional, concurrent, and object-oriented programming languages.

We have pursued our work on session types along three main directions.

Multiparty Reactive Sessions

Ensuring that communication-centric systems interact according to an intended protocol is an important but difficult problem, particularly for systems with some reactive or timed components. To rise to this challenge, we have studied the integration of session-based concurrency and Synchronous Reactive Programming (SRP).

Synchronous reactive programming (SRP) is a well-established programming paradigm whose essential features are logical instants, broadcast events and event-based preemption. This makes it an ideal vehicle for the specification and analysis of reactive systems. *Session-based concurrency* is the model of concurrent computation induced by session types, a rich typing discipline designed to specify the structure of interactions.

In this work, we propose a process calculus for multiparty sessions enriched with features from SRP. In this calculus, protocol participants may broadcast messages, suspend themselves while waiting for a message, and also react to events.

Our main contribution is a session type system for this calculus, which enforces session correctness in terms of communication safety and protocol fidelity, and also ensures a time-related property, which we call input timeliness, which entails livelock-freedom. Our type system departs significantly from existing ones, specifically as it captures the notion of "logical instant" typical of SRP. This work is currently under submission.

Reversible Sessions with Flexible Choices

Reversibility has been an active trend of research for the last fifteen years. A reversible computation is a computation that has the ability to roll back to a past state. Allowing computations to reverse is a means to improve system flexibility and reliability. In the setting of concurrent process calculi, reversible computations have been first studied for CCS, then for the π-calculus, and only recently for session calculi.

Following up on our previous work on concurrent reversible sessions [29], we studied a simpler but somewhat "more realistic" calculus for concurrent reversible multiparty sessions, equipped with a flexible choice operator allowing for different sets of participants in each branch of the choice. This operator is inspired by the notion of *connecting action* recently introduced by Hu and Yoshida to describe protocols with optional participants. We argue that this choice operator allows for a natural description of typical communication protocols. Our calculus also supports a compact representation of the history of processes and types, which facilitates the definition of rollback. Moreover, it implements a fine-tuned strategy for backward computation. We present a session type system for the calculus and show that it enforces the expected properties of session fidelity, forward progress and backward progress. This work has been accepted for journal publication.

Multiparty sessions with Internal Delegation

We have investigated a new form of delegation for multiparty session calculi. Usually, delegation allows a session participant to appoint a participant in another session to act on her behalf. This means that delegation is inherently an inter-session mechanism, which requires session interleaving. Hence delegation falls outside the descriptive power of global types, which specify single sessions. As a consequence, properties such as deadlock-freedom or lock-freedom are difficult to ensure in the presence of delegation. Here we adopt a different view of delegation, by allowing participants to delegate tasks to each other within the same multiparty session. This way, delegation occurs within a single session (internal delegation) and may be captured by its global type. To increase flexibility in the use of delegation, our calculus uses connecting communications, which allow optional participants in the branches of choices. By these means, we are able to express conditional delegation. We present a session type system based on global types with internal delegation, and show that it ensures the usual safety properties of multiparty sessions, together with a progress property. This work is under submission.

5.5. Measurement and Detection of Web Tracking

Detecting Web Trackers via Analyzing Invisible Pixels

The Web has become an essential part of our lives: billions are using Web applications on a daily basis and while doing so, are placing *digital traces* on millions of websites. Such traces allow advertising companies, as well as data brokers to continuously profit from collecting a vast amount of data associated to the users.

Web tracking has been extensively studied over the last decade. To detect tracking, most of the research studies and user tools rely on *consumer protection lists*. EasyList [23] and EasyPrivacy [24] (EL&EP) are the most popular publicly maintained blacklist of know advertising and tracking domains, used by the popular browser extensions AdBlock Plus [20] and uBlockOrigin [28]. Disconnect [22] is another very popular list for detecting domains known for tracking, used in Disconnect browser extension [21] and in integrated tracking protection of Firefox browser [25]. Relying on EL&EP or Disconnect became the *de facto* approach to detect third-party tracking requests in privacy and measurement community. However it is well-known that these lists detect only known tracking and ad-related requests, and a tracker can easily avoid this detection by registering a new domain or changing the parameters of the request.

In this work, to detect trackers, we propose a new technique based on the analysis of invisible pixels [2]. These images are routinely used by trackers in order to send information or third-party cookies back to their servers: the simplest way to do it is to create a URL containing useful information, and to dynamically add an image HTML tag into a webpage. Since invisible pixels do not provide any useful functionality, we consider them *perfect suspects for tracking*.

By using an Inria cluster and setting up a distributed crawler, we have collected a dataset of invisible pixels from 829,349 webpages. By analyzing this dataset, we observed that invisible pixels are widely used: more than 83% of pages incorporate at least one invisible pixel.

Overall, we made the following key contributions:

- We define a new classification of Web tracking behaviors based on the analysis of invisible pixels. By analyzing behavior associated to the delivery of invisible pixels, we propose a new fine-grained classification of tracking behaviors, that consists of 8 categories of tracking. To our knowledge, *we are the first to analyse tracking behavior based on invisible pixels that are present on 83% of the webpages*.
- We apply our classification to a full dataset and uncover new collaborations between third-party domains. We detect new relationships between third-party domains beyond basic cookie syncing detected in the past. In particular, we discovered that *first to third party cookie syncing* is the most prevalent tracking behavior performed by 50,812 distinct domains. Finally, we find that 76.23% of requests responsible for tracking originate from loading other resources than invisible images. To our knowledge, *we are the first to discover a highly prevalent first to third party syncing behavior detected on 51.54% of all crawled domains*.
- We show that the consumer protection lists cannot be considered as ground truth to identify trackers. We find out that the browser extensions based on EasyList and EasyPrivacy (EL&EP) and Disconnect each miss 22% of tracking requests we detect. Moreover, if we combine all the lists, 238,439 requests originated from 7,773 domains are unknown to these lists and hence still track users on 5,098 webpages even if tracking protection is installed. We also detect instances of cookie syncing in domains unknown to these lists and therefore likely unrelated to advertising. To our knowledge, *we are the first to detect that EL&EP and also Disconnect lists used in majority of Web Tracking detection literature are actually missing tracking requests to 7,773 distinct domains.*

This working paper [19] is currently under submission at an international conference.

A survey on Browser Fingerprinting

This year, we have conducted a survey on the research performed in the domain of browser fingerprinting, while providing an accessible entry point to newcomers in the field. We explain how this technique works and where it stems from. We analyze the related work in detail to understand the composition of modern fingerprints and see how this technique is currently used online. We systematize existing defense solutions into different categories and detail the current challenges yet to overcome.

[2] By “invisible pixels” we mean 1x1 pixel images or images without content.

A *browser fingerprint* is a set of information related to a user's device from the hardware to the operating system to the browser and its configuration. *Browser fingerprinting* refers to the process of collecting information through a web browser to build a fingerprint of a device. Via a script running inside a browser, a server can collect a wide variety of information from public interfaces called Application Programming Interface (API) and HTTP headers. An API is an interface that provides an entry point to specific objects and functions. While some APIs require a permission to be accessed like the microphone or the camera, most of them are freely accessible from any JavaScript script rendering the information collection trivial. Contrarily to other identification techniques like cookies that rely on a unique identifier (ID) directly stored inside the browser, browser fingerprinting is qualified as completely *stateless*. It does not leave any trace as it does not require the storage of information inside the browser.

The goal of this work is twofold: first, to provide an accessible entry point for newcomers by systematizing existing work, and second, to form the foundations for future research in the domain by eliciting the current challenges yet to overcome. We accomplish these goals with the following contributions:

- A thorough survey of the research conducted in the domain of browser fingerprinting with a summary of the framework used to evaluate the uniqueness of browser fingerprints and their adoption on the web.
- An overview of how this technique is currently used in both research and industry.
- A taxonomy that classifies existing defense mechanisms into different categories, providing a high-level view of the benefits and drawbacks of each of these techniques.
- A discussion about the current state of browser fingerprinting and the challenges it is currently facing on the science, technological, business, and legislative aspects.

This work has been submitted for publication at an international journal.

Measuring Uniqueness of Browser Extensions and Web Logins

Web browser is the tool people use to navigate through the Web, and privacy research community has studied various forms of *browser fingerprinting*. Researchers have shown that a user's browser has a number of inherent "physical" characteristics that can be used to uniquely identify her browser and hence to track it across the Web. Fingerprinting of users' devices is similar to physical biometric traits of people, where only physical characteristics are studied.

Similar to previous demonstrations of user uniqueness based on their behavior, *behavioral characteristics*, such as browser settings and the way people use their browsers can also help to uniquely identify Web users. For example, a user installs web browser extensions she prefers, such as AdBlock, LastPass, or Ghostery to enrich her Web experience. Also, while browsing the Web, she logs in her preferred social networks, such as Gmail, Facebook or LinkedIn. In this work, we study *users' uniqueness* based on their behavior and preferences on the Web: we analyze how unique are Web users based on their *browser extensions and logins*.

In this work, we performed the first large-scale study of user uniqueness based on browser extensions and Web logins, collected from more than 16,000 users who visited our website https://extensions.inrialpes.fr/. Our experimental website identifies installed Google Chrome extensions via Web Accessible Resources. and detects websites where the user is logged in by methods that rely on URL redirection and CSP violation reports. Our website is able to detect the presence of 13K Chrome extensions (the number of detected extensions varied monthly between $12,164$ and $13,931$), covering approximately 28% of all free Chrome extensions [3] . We also detect whether the user is connected to one or more of 60 different websites. Our main contributions are:

- A large scale study on *how unique users are based on their browser extensions and website logins*. We discovered that 54.86% of users that have installed at least one detectable extension are unique; 19.53% of users are unique among those who have logged into one or more detectable websites; and 89.23% are unique among users with at least one extension and one login. Moreover, we discover that 22.98% of users could be uniquely identified by web logins, even if they disable JavaScript.

[3] The list of detected extensions and websites are available on our website: https://extensions.inrialpes.fr/faq.php

- We study the privacy dilemma on Adblock and privacy extensions, that is, *how well these extensions protect their users against trackers and how they also contribute to uniqueness*. We evaluate the statement “the more privacy extensions you install, the more unique you are” by analyzing how users’ uniqueness increases with the number of privacy extensions she installs; and by evaluating the tradeoff between the privacy gain of the blocking extensions such as Ghostery [26] and Privacy Badger [27].

We furthermore show that browser extensions and web logins can be exploited to fingerprint and track users by only checking a limited number of extensions and web logins. We have applied an advanced fingerprinting algorithm [30] that carefully selects a limited number of extensions and logins. For example, we show that 54.86% of users are unique based on all 16,743 detectable extensions. However, by testing 485 carefully chosen extensions we can identify more than 53.96% of users. Besides, detecting 485 extensions takes only 625ms.

Finally, we give suggestions to the end users as well as website owners and browser vendors on how to protect the users from the fingerprinting based on extensions and logins.

This paper has been published at at WPES international workshop affiliated with ACM CCS 2018 [14].

6. Bilateral Contracts and Grants with Industry

6.1. Bilateral Grants with Industry

The ANSWER project (Advanced aNd Secured Web Experience and seaRch) is lead by the QWANT search engine and the Inria Sophia Antipolis Méditerranée research center. This proposal is the winner of the "Grand Challenges du Numérique" (BPI) and aims to develop the new version of the search engine http://www.qwant.com with radical innovations in terms of search criteria, indexed content and privacy of users. The project started on January 1, 2018. In the context of this project, we got

- with Arnaud Legout from the DIANA project-team a funding for a 3 years Ph.D. student to work on Web tracking technologies and privacy protection. Imane Fouad was hired to work on this project.
- a funding for 18 months Postdoc to work on Web application security. Yoonseok Ko was hired to work on this project as a postdoc.

7. Partnerships and Cooperations

7.1. Inria internal funding

7.1.1. IPL SPAI

SPAI (Security Program Analyses for the IoT) is an IPL (Inria Project Lab), with a duration of 4 years, started on April 2018. Members of the Antique, Celtique, Indes,Kairos, and Privatics Inria teams are involved in the SPAI IPL.

SPAI is concerned with the design of program analyses for a multitier language for the Internet of Things (IoT). The programming abstractions will allow us to reason about IoT systems from microcontrollers to the cloud. Relying on the Inria multitier language Hop.js semantics and the current Coq formalizations of JavaScript semantics, we plan to certify these analyses in order to guarantee the impossibility of security properties violations and implement security properties’ enforcements by compilation.

Tamara Rezk coordinates this project.

7.1.2. ADT FingerKit

In the context of the Inria ADT call, we are involved in a *FingerKit: a Cloud Platform to Study Browser Fingerprints at Large*, lead by Walter Rudametkin from the Spirals project-team. The funding for a two year engineering position for the 2018-2020 period was obtained and an engineer is hired in Spirals project-team. Nataliia Bielova from INDES team is part of this project.

7.2. National Initiatives

7.2.1. ANR AJACS

The AJACS project (Analyses of JavaScript Applications: Certification & Security) is funded by the ANR for 42 months, starting December 2014. The goal of the AJACS project is to provide strong security and privacy guarantees on the client side for web application scripts. The Indes members Tamara Rezk and Nataliia Bielova are involved in the tasks WP2 Certified Analyses and WP3 Security of JavaScript Applications. The partners of this project include Inria teams Celtique (coordinator), Toccata, and Prosecco.

7.2.2. ANR CISC

The CISC project (Certified IoT Secure Compilation) is funded by the ANR for 42 months, starting in April 2018. The goal of the CISC project is to provide strong security and privacy guarantees for IoT applications by means of a language to orchestrate IoT applicatoins from the microcontroller to the cloud. Tamara Rezk coordinates this project, and Manuel Serrano, Ilaria Castellani and Nataliia Bielova participate in the project. The partners of this project are Inria teams Celtique, Indes and Privatics, and Collège de France.

7.2.3. ANR PrivaWeb

The PrivaWeb project (Privacy Protection and ePrivacy Compliance for Web Users) is funded by the ANR JCJC program for 42 months, starting in December 2018. PrivaWeb aims at developing new methods for detection of new Web tracking technologies and new tools to integrate in existing Web applications that seamlessly protect privacy of users. Nataliia Bielova coordinates this project.

7.2.4. FUI UCF

The 3 years long UCF project aims at developing a reactive Web platforms for delivering multimedia contents. The partners of the project are the startups Alterway, OCamlPro, and XWiki, and the academic research laboratories of University Pierre et Marie Curie, and Denis Diderot. Manuel Serrano participates in this project.

7.3. European Initiatives

7.3.1. Collaborations in European Programs, Except FP7 & H2020

7.3.1.1. ICT Cost Action IC1405 on Reversible Computation

Program: ICT COST Action IC1405

Project title: Reversible computation - extending horizons of computing

Duration: November 2014 - April 2019

Coordinator: Irek Ulidowski, University of Leicester

Other partners: several research groups, belonging to 23 European countries.

Abstract: Reversible computation is an emerging paradigm that extends the standard mode of computation with the ability to execute in reverse. It aims to deliver novel computing devices and software, and to enhance traditional systems. The potential benefits include the design of reversible logic gates and circuits - leading to low-power computing and innovative hardware for green ICT, new conceptual frameworks and language abstractions, and software tools for reliable and recovery-oriented distributed systems. This is the first European network of excellence aimed at coordinating research on reversible computation.

7.3.1.2. Bilateral PICS project SuCCeSS

Program: CNRS Bilaterial PICS project

Project acronym: SuCCeSS

Project title: Security, Adaptability and time in Communication Centric Software Systems

Duration: June 2016 - June 2019

Coordinator: Cinzia Di Giusto, I3S, Sophia Antipolis

Partners: I3S, Inria, University of Groningen

Abstract: The project SuCCeSS is a CNRS-funded “Projet coopératif” (PICS 07313), involving two French teams in Sophia Antipolis (the MDSC team at the laboratory I3S, acting as coordinator, and the INDES team) and one Dutch team at the University of Groningen. The project started in June 2016 and is due to end in June 2019. The objective of the project is to study formal models for reliable distributed communication-centric software systems. The project focusses on analysis and validation techniques based on behavioural types, aimed at enforcing various properties (safety, liveness, security) of structured communications.

7.4. International Initiatives

7.4.1. Inria International Partners

7.4.1.1. Informal International Partners

- We are initiating a new collaboration with Prof. Robby Findler and his group from Northwestern University of Chicago. We are studying reactive synchronous programming languages and their applications.
- We are pursuing our collaboration on session types with Prof. Mariangiola Dezani Ciancaglini from the University of Torino and Prof. Paola Giannini from the University of Piemonte Orientale. We also continue to collaborate with Dr. Jorge Pérez and his PhD student Mauricio Cano, from the University of Groningen, on the integration of session types with synchronous reactive programming.
- We are initiating a new collaboration with Professor of Law, Frederik Zuiderveen Borgesius from the Radbound University Nijmegen and Amsterdam Law School (double affiliation). We are studying General Data Protection Regulation (GDPR) and ePrivacy Regulation and their application to Web tracking technologies.
- We have been collaborating with Prof. Alejandro Russo from Chalmers University of Technology and Prof. Cormac Flanagan from University of California Santa Cruz, that resulted in a joint publication at WWW conference [15].
- We have been collaborating with Prof. Benoit Baudry from KTH Royal Institute of Technology, Sweden on the survey of browser fingerprinting technologies.

7.4.2. Participation in Other International Programs

7.4.2.1. International Initiatives

DAJA

Title: Detection strategies based on Software Metrics for Multitier JavaScript

International Partners (Institution - Laboratory - Researcher):

Universidad de Chile (Chile), DDC Alexandre Bergel

Universidad Nacional del Centro de la Provincia de Buenos Aires (Argentina) - ISISTAN Research Insitute - Santiago Vidal

Duration: 2018 - 2019

Start year: 2018

See also: https://daja-sticamsud.github.io/

JavaScript is the most popular object scripting programming language. It is extensively used conceived only for scripting, it is frequently used in large applications. The rapid adoption of JavaScript has outpaced the Software Engineering community to propose solutions to ensure a satisfactory code quality production. This situation has favored the production of poor quality JavaScript applications: we have found across JavaScript applications a large presence of dead-code (i.e., source code portion that is never used) and code duplications. These symptoms are known to lead to maintenance and performance degradation. Moreover, we have previously analyzed potential security threats to JavaScript applications produced by bad coding practices.

The DAJA project will provide methodologies, techniques, and tools to ease the maintenance of software applications written in JavaScript while improving its security.

7.5. International Research Visitors

7.5.1. Visits of International Scientists

- We are collaborating with Prof. Marc Feeley from University of Montréal. For the second consecutive year, M. Feeley has visited us for studying implementation of dynamic languages.
- As part of our ongoing collaboration on session types, Prof. Paola Giannini from the University of Piemonte Orientale visited our team for two weeks, funded by the COST Action on Reversibility.
- Our team, together with Cinzia Di Giusto's team at I3S, hosted Mauricio Cano, a PhD student from the University of Groningen, for a 2-month visit. This was part of our collaboration with the University of Groningen within the project PICS SuCCeSS. The visit was funded for the most part by Academy 1 of Université Côte d'Azur.

7.5.1.1. Internships

- Tamara Rezk supervised the intern Sadry Fievet for 6 months
- Tamara Rezk supervised - as "tuteur" - the internship of El Mehdi Regragui for 6 months
- Bertrand Petit and Manuel Serrano supervised the internship of Thibaud Ardoin who studied and implemented the Skini distributed sequencer.

8. Dissemination

8.1. Promoting Scientific Activities

8.1.1. Scientific Events Organisation

8.1.1.1. General Chair, Scientific Chair

- Nataliia Bielova was the co-chair (together with Claude Castelluccia) of the Francophone workshop on Privacy Protection “ l'Atelier sur la Protection de la Vie Privée” (APVP), which took place in Porquerolles (France) from 3 to 6 June 2018. https://project.inria.fr/apvp2018/
- Manuel Serrano was the general chair of the Programming'18 conference that took place in Nice in April 2018, https://2018.programmingconference.org/.
- Tamara Rezk was co-chair and co-organizer (together with Sébastien Bardin and Stéphanie Delaune) of the yearly event of the Working Group “Méthodes formelles pour la sécurité” (GT-MFS) in the context of the Pré-GDR Sécurité meeting in Paris (May 30, 2018).
- Ilaria Castellani was the co-chair (together with Mohammad Reza Mousavi) of the workshop TRENDS 2018, which took place in Beijing on September 8, in association with the CONCUR 2018 conference. https://concurrency-theory.org/events/workshops/trends

8.1.1.2. Member of the Organizing Committees

- Tamara Rezk was the local chair of the Programming'18 conference that took place in Nice in April 2018 https://2018.programmingconference.org/.

8.1.2. Scientific Events Selection

8.1.2.1. Chair of Conference Program Committees

- Nataliia Bielova was the PC co-chair (together with Claude Castelluccia) of the Francophone workshop on Privacy Protection " l'Atelier sur la Protection de la Vie Privée" (APVP), which tool place in Porquerolles (France) from 3 to 6 June 2018. https://project.inria.fr/apvp2018/
- Manuel Serrano was the co-chair together with Sukyong Ryu from Kaist University of the WWW'18 alternate Programming Track, https://www2018.thewebconf.org/call-for-papers/web-programming-cfp/, which took place in April in Lyon.

8.1.2.2. Member of the Conference Program Committees

- Nataliia Bielova served in the Program Committees of IEEE SecDev'18, POST'18, ProWeb'18, eCrime'18, APVP'18. She also is a member of the Steering Committee of the PLAS workshop.
- Manuel Serrano served in the Program Committee of ProWeb'18 workshop.
- Tamara Rezk served in the Program Committees of NDSS'18, Euro S&P'18, APLAS'18, POST'18, WWW'18, PriSC'18, CSF'18, SEC@SAC'18. She is also a member of the Steering Committee of the POST conference at ETAPS.
- Ilaria Castellani served in the Program Committee of the conference FORTE'18.

8.1.2.3. Reviewer

- The team members have been reviewers for the following conferences and workshops: PoPETs'18, CONCUR'18, FORTE'18, EXPRESS/SOS'18.

8.1.3. Journal

8.1.3.1. Member of the Editorial Boards

- Ilaria Castellani is a member of the editorial board of *Technique et Science Informatiques*.
- Tamara Rezk is a member of editorial Board of Interstices and blog Binaire du Le Monde.

8.1.3.2. Reviewer - Reviewing Activities

- The team members have been reviewers for the following journals: JCS, Acta Informatica, JLAMP, TCS, JOT.

8.1.4. Invited Talks

- Nataliia Bielova was an invited speaker at the workshop "Transparence et opacité des systèmes d'information" in April http://transparence.conf.citi-lab.fr/. She gave a talk at the Journées Scientifiques Inria in June https://journees-scientifiques2018.inria.fr/francais-programme/. She gave a tutorial on Web Tracking Technologies at TheWeb'18 conference in April https://www2018.thewebconf.org/program/tutorials-track/tutorial-203/ and gave a keynote talk at the Journée scientifique RISE/DS4H of Université Cote d'Azur in September. Nataliia was an invited speaker at Harvard University where she gave a talk on Web tracking technologies in October 2018. She also gave an invited lecture on "Online tracking and privacy protection", at Master 2 course, Carnegie Mellon University in October and an invited lecture on "Ethics and Internet Security" in Master in Digital Business course at SKEMA Business school in March.
- Manuel Serrano gave an invited talk on on Web Programming at the Programming ProWeb'18 workshop in April, https://2018.programming-conference.org/track/proweb-2018-papers. He gave a two hours talk on Reactive Web Programming during the Synchron'18 workshop https://project.inria.fr/synchron2018/fr/. He also was an invited speaker at Brown University where he gave a talk on JavaScript implementation.

- Tamara Rezk was an invited speaker at the seminar in the context of Inaugural lecture of Marieke Huisman, University of Twente, January 2018. She also gave a talk in the Dagstuhl Seminar on Secure Compilation of May 2018 and at Harvard University in October 2018.

8.1.5. Leadership within the Scientific Community

- Ilaria Castellani is the chair of the IFIP TC1 WG 1.8 on Concurrency Theory since June 2014. In 2018 she has been reelected for a second term.
- Ilaria Castellani is a Management Committee member of the COST Action IC1405 on Reversible Computation (November 2014-April 2019).

8.1.6. Research Administration

- Nataliia Bielova is a member of “Comité du Suivi Doctoral (CSD)” (Supervision of PhD students) of the Inria Sophia Antipolis Mediterranée research center. She was a member of the Inria “Master Transverse” evaluation committee.
- Manuel Serrano was the "Délégué scientifique" (head of research) of the Inria Sophia Antipolis Mediterranée research center until end of August. He was member of the Inria Evaluation Committee until that time.
- Tamara Rezk is a member of " Commission de Développement Technologique (CDT)" of the Inria Sophia Antipolis Mediterranée research center.
- Ilaria Castellani is a member of Inria’s “Comité Parité et Égalité des Chances”. In the Centre of Inria Sophia Antipolis, she is a member of the “Comité d’Animation et Médiation Scientifique” and of the Scientific Committee of the Morgenstern Colloquium. She is also a member of the Réseau Parité of Université Côte d’Azur, and of the Scientific Committee of the Forum Numerica Seminar of Academy 1 of Université Côte d’Azur.

8.2. Teaching - Supervision - Juries

8.2.1. Teaching

Master : Nataliia Bielova, Security of Web Applications, 18ETD, niveau M2, University of Pierre et Marie Curie, France

Master : Nataliia Bielova, Foundations of Privacy, 1.5ETD, niveau M2, Carnegie Mellon University, USA

Master: Nataliia Bielova, Ethics and Internet Security, 1.5ETD, M2, SKEMA Business school, France.

Master : Tamara Rezk, Security of Web Applications, 28ETD, niveau M2, University of Nice Sophia Antipolis, France

Master : Tamara Rezk, Preuves en Cryptographie, 28ETD, niveau M2, University of Nice Sophia Antipolis, France

Master : Tamara Rezk, Security of Web Applications, 36ETD, niveau M2, University of Pierre et Marie Curie, France

Doctorat : Nataliia Bielova, Apprentissage et fouille de données sur les Réseaux, 2.25ETD, École de Recherche ResCom, France

Doctorat : Nataliia Bielova, Ecole de Cybersécurité, 3ETD, UCA, France

Doctorat : Nataliia Bielova, Web Privacy, 30ETD, University of Trento, Italy

E-learning

> MOOC: Nataliia Bielova, “Protection de la vie privée dans le monde numérique”, 1 Module, 8 sequences, FUN-MOOC, Inria, grande public, formation continue, 8900 inscrit au 18 Decembre 2018.

8.2.2. Supervision

HdR :Tamara Rezk, *Secure Programming* [11], Université Côte d'Azur, 03/04/2018

PhD : Dolière Francis Somé, *Web Applications Security and Privacy* [12], Université Cote d'Azur, 29/10/2018, Nataliia Bielova and Tamara Rezk

PhD: Colin Vidal, *Programmation Web réactive* [13], University of Nice, 1/07/2015-1/09/2018, Manuel Serrano and Gérard Berry.

PhD in progress : Imane Fouad, Web tracking detection and measurement, 1/01/2018, Nataliia Bielova and Arnaud Legout

PhD in progress : Jayanth Krishnamurthy, Privacy policy enforcement in IoT applications, 12/09/2018, Nataliia Bielova and Manuel Serrano

PhD in progress : Héloïse Maurel, Secure compilation of IoT applications, 1/10/2018, Tamara Rezk

PhD in progress : Mohamad Ellaz, Encodings of ElGammal, 1/12/2017, Benjamin Gregoire and Tamara Rezk

PhD in progress : Lesly-Ann Daniel, Security analysis of binary code, 1/10/2018, Sébastien Bardin and Tamara Rezk

Postdoc: Yoonseok Ko, Subsets of secure JavaScript, 1/10/2018-, Tamara Rezk

Postdoc: Minh Ngo, TSNI enforcement mechanisms, 1/01/2018-31/12/2018, Tamara Rezk

Postdoc: Francis Somé, IoT secure broadcasting, 1/11/2018-, Tamara Rezk

8.2.3. Juries

- Nataliia Bielova was a member of the PhD jury of Oleksii Starov, Stony Brook University.
- Nataliia Bielova was a member of the PhD jury of Daniel Schoepe, Chalmers University of Technology.
- Manuel Serrano was a member of the PhD jury of Remy El-Sibaie, Paris-Sorbonne University.
- Manuel Serrano was a member of the PhD jury of Pierre Talbot Ircam, Paris-Sorbonne University.
- Tamara Rezk was "rapporteur" for the PhD thesis of Nadim Kobeissi, ENS
- Tamara Rezk was "rapporteur" for the PhD thesis of Gurvan Cabon, Université de Rennes
- Tamara Rezk was a member of the Licenciate jury of Iulia Bastys, Chalmers University of Technology
- Tamara Rezk was a member of the jury "soutenance stage" of the CASPAR master of University of Nice Sophia Antipolis
- Tamara Rezk was a member of the jury for the ACM Student Research Competition of April 2018

8.3. Popularization

8.3.1. Internal or external Inria responsibilities

Tamara Rezk is member of the editorial board of Interstices and Blog Binaire Le Monde

8.3.2. Articles and contents

Tamara Rezk published the article "Est-ce que mon programme est bien protégé ?" in blog binaire Le Monde in February 2018. The article with some minor modifications was retaken by Interstices and published in December 2018. http://binaire.blog.lemonde.fr/2018/02/09/est-ce-que-mon-programme-est-bien-protege-contre-les-cyberattaques/.

Nataliia Bielova was interviewed by Sophie Casals, "Comment mieux protéger sa vie privée en naviguant sur internet? Les conseils d'une spécialiste" in Nice Matin, June 2018. https://www.nicematin.com/faits-de-societe/comment-mieux-proteger-sa-vie-privee-en-naviguant-sur-internet-les-conseils-dune-specialiste-239166

8.3.3. Education

8.3.3.1. Skini and the Fabrique à Musique

Skini is currently used with a class of secondary school in order to create a musical show in collaboration with the Conservatory of Nice and the CIRM (National Center for Music Creation). A group of 25 children, 12 year old, are creating together a story, and use the distributed sequencer of Skini to design basic musical patterns using tablets. The next step is to organize with them the orchestration of the piece which is coded in HipHop. The activation of the piece will be done in real time by the children using the tablet and the Skini plateform. This project is funded by SACEM (Société des auteurs, compositeurs et éditeurs de musique). The project will end in May 2019 with a public concert in the Conservatory of Nice.

8.3.3.2. MOOC Protection de la vie privée dans le monde numérique

Nataliia Bielova has participated to the MOOC “Protection de la vie privée dans le monde numérique” (Session 2) that was shooted in July 2018 under the guidance of Inria Learning Lab and was open to general public between 5 November 2018 and 7 January 2019, https://www.fun-mooc.fr/courses/course-v1:inria+41015+session02/about.

8.3.4. Interventions

- Nataliia Bielova held a stand at Portes ouvertes (Fête de la science) of Inria Sophia Antipolis explaining how to protect privacy on the Web, October 2018.
- Ilaria Castellani and Héloïse Maurel held the stand “Science au féminin” together with Carine Antico from CNRS at the Fête de la Science in Antibes Juan-les-Pins, October 2018.

8.3.5. Internal action

- Nataliia Bielova gave a talk at the PhD seminar special edition on “Tracking technologies and protection of your privacy on the Web”, March 2018.
- Nataliia Bielova gave a talk at the Data Science meetup Nice on “How companies track you as you browse the web and how to protect yourself?”, https://www.meetup.com/Data-Science-Meetup-Nice-Sophia-Antipolis/events/255546602/, October 2018.

9. Bibliography

Major publications by the team in recent years

[1] N. Bielova, T. Rezk. *A Taxonomy of Information Flow Monitors*, in "International Conference on Principles of Security and Trust (POST 2016)", Eindhoven, Netherlands, F. Piessens, L. Viganò (editors), LNCS - Lecture Notes in Computer Science, Springer, April 2016, vol. 9635, pp. 46–67 [*DOI :* 10.1007/978-3-662-49635-0_3], https://hal.inria.fr/hal-01348188

[2] G. Boudol, I. Castellani. *Noninterference for Concurrent Programs and Thread Systems*, in "Theoretical Computer Science", 2002, vol. 281, n^{o} 1, pp. 109-130

[3] G. Boudol, Z. Luo, T. Rezk, M. Serrano. *Reasoning about Web Applications: An Operational Semantics for HOP*, in "ACM Transactions on Programming Languages and Systems (TOPLAS)", 2012, vol. 34, n^{o} 2

[4] S. Capecchi, I. Castellani, M. Dezani-Ciancaglini. *Information Flow Safety in Multiparty Sessions*, in "Mathematical Structures in Computer Science", 2015, vol. 26, n^{o} 8, 43 p. [*DOI :* 10.1017/S0960129514000619], https://hal.inria.fr/hal-01237236

[5] I. Castellani, M. Dezani-Ciancaglini, P. Giannini. *Concurrent Reversible Sessions*, in "CONCUR 2017 - 28th International Conference on Concurrency Theory ", Berlin, Germany, CONCUR 2017, Roland Meyer and Uwe Nestmann, September 2017, vol. 85, pp. 1-17 [*DOI :* 10.4230/LIPIcs.CONCUR.2017.30], https://hal.inria.fr/hal-01639845

[6] C. Fournet, T. Rezk. *Cryptographically sound implementations for typed information-flow security*, in "Proceedings of the 35th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2008, San Francisco, California, USA, January 7-12, 2008", 2008, pp. 323-335

[7] M. Ngo, F. Piessens, T. Rezk. *Impossibility of Precise and Sound Termination-Sensitive Security Enforcements*, in "SP 2018 - IEEE Symposium on Security and Privacy", San Francisco, United States, IEEE, May 2018, pp. 496-513 [*DOI :* 10.1109/SP.2018.00048], https://hal.inria.fr/hal-01928669

[8] M. Serrano, G. Berry. *Multitier Programming in Hop - A first step toward programming 21st-century applications*, in "Communications of the ACM", August 2012, vol. 55, n^{o} 8, pp. 53–59 [*DOI :* 10.1145/2240236.2240253], http://cacm.acm.org/magazines/2012/8/153796-multitier-programming-in-hop/abstract

[9] M. Serrano, V. Prunet. *A Glimpse of Hopjs*, in "21th ACM Sigplan Int'l Conference on Functional Programming (ICFP)", Nara, Japan, September 2016, pp. 188–200, http://dx.doi.org/10.1145/2951913.2951916

[10] D. F. Somé, N. Bielova, T. Rezk. *On the Content Security Policy Violations due to the Same-Origin Policy*, in " 26th International World Wide Web Conference, 2017 (WWW 2017)", April 2017 [*DOI :* 10.1145/3038912.3052634], https://hal.inria.fr/hal-01649526

Publications of the year

Doctoral Dissertations and Habilitation Theses

[11] T. Rezk. *Secure Programming*, Université de Nice - Sophia Antipolis, April 2018, Habilitation à diriger des recherches, https://hal.inria.fr/tel-01941697

[12] D. F. Somé. *Web applications Security and Privacy*, Université Côte D'Azur, October 2018, https://hal.inria.fr/tel-01925851

[13] C. Vidal. *Reactive Web Programming*, Université Côte d'Azur, July 2018, https://tel.archives-ouvertes.fr/tel-01900619

International Conferences with Proceedings

[14] G. G. Gulyás, D. F. Somé, N. Bielova, C. Castelluccia. *To Extend or not to Extend: On the Uniqueness of Browser Extensions and Web Logins*, in "WPES'18 - Workshop on Privacy in the Electronic Society", Toronto, Canada, ACM Press, October 2018, pp. 14-27 [*DOI :* 10.1145/3267323.3268959], https://hal.inria.fr/hal-01921863

[15] M. Ngo, N. Bielova, C. Flanagan, T. Rezk, A. Russo, T. Schmitz. *A Better Facet of Dynamic Information Flow Control*, in "WWW '18 Companion: The 2018 Web Conference Companion", Lyon, France, April 2018, pp. 1-9, https://hal.inria.fr/hal-01723723

[16] M. NGO, F. PIESSENS, T. REZK. *Impossibility of Precise and Sound Termination-Sensitive Security Enforcements*, in "SP 2018 - IEEE Symposium on Security and Privacy", San Francisco, United States, IEEE, May 2018, pp. 496-513 [*DOI :* 10.1109/SP.2018.00048], https://hal.inria.fr/hal-01928669

[17] M. SERRANO. *JavaScript AOT compilation*, in "the 14th ACM SIGPLAN International Symposium on Dynamic Languages", Boston, France, Proceeding of the 14th ACM SIGPLAN International Symposium on Dynamic Languages, ACM Press, November 2018 [*DOI :* 10.1145/3276945.3276950], https://hal.archives-ouvertes.fr/hal-01937197

[18] C. VIDAL, G. BERRY, M. SERRANO. *Hiphop.js: a language to orchestrate web applications*, in "SAC: Symposium on Applied Computing", Pau, France, April 2018, vol. Proceedings of the 2018 Symposium on Applied Computing [*DOI :* 10.1145/3167132.3167440], https://hal.archives-ouvertes.fr/hal-01937252

Other Publications

[19] I. FOUAD, N. BIELOVA, A. LEGOUT, N. SARAFIJANOVIC-DJUKIC. *Tracking the Pixels: Detecting Web Trackers via Analyzing Invisible Pixels*, December 2018, working paper or preprint, https://hal.inria.fr/hal-01943496

References in notes

[20] *Adblock Plus Official website*, 2018, https://adblockplus.org/

[21] *Disconnect Official website*, 2018, https://disconnect.me/

[22] *Disconnect List*, 2018, https://disconnect.me/trackerprotection/blocked

[23] *EasyList filter lists*, 2018, https://easylist.to/

[24] *EasyPrivacy filter lists*, 2018, https://easylist.to/easylist/easyprivacy.txt

[25] *The new Firefox Fast for good*, 2018, https://www.mozilla.org/en-US/firefox/new/

[26] *Ghostery Official website*, 2018, https://www.ghostery.com/

[27] *Privacy Badger Official website - Electronic Frontier Foundation*, 2018, https://www.eff.org/privacybadger

[28] *uBlock Origin - An efficient blocker for Chromium and Firefox. Fast and lean*, 2018, https://github.com/gorhill/uBlock

[29] I. CASTELLANI, M. DEZANI-CIANCAGLINI, P. GIANNINI. *Concurrent Reversible Sessions*, in "CONCUR 2017 - 28th International Conference on Concurrency Theory ", Berlin, Germany, CONCUR 2017, Roland Meyer and Uwe Nestmann, September 2017, vol. 85, pp. 1-17 [*DOI :* 10.4230/LIPIcs.CONCUR.2017.30], https://hal.inria.fr/hal-01639845

[30] G. G. GULYÁS, G. ACS, C. CASTELLUCCIA. *Near-Optimal Fingerprinting with Constraints*, in "Proceedings on Privacy Enhancing Technologies", 2016, vol. 2016, n^{o} 4, pp. 470–487