Activity Report 2019

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]
Gérard Berry [Collège de France, Senior Researcher, HDR]

Post-Doctoral Fellows

Yoon Seok Ko [Inria, Post-Doctoral Fellow]
Celestin Matte [Inria, Post-Doctoral Fellow, from Mar 2019]
Doliere Some [Inria, Post-Doctoral Fellow, until Apr 2019]

PhD Students

Feras Al Kassar [Inria, PhD Student, until Sep 2019]
Lesly Ann Daniel [CEA, PhD Student]
Imane Fouad [Inria, PhD Student]

Jayanth Krishnamurthy [Inria, PhD Student]
Héloïse Maurel [Inria, PhD Student]
Bertrand Petit [Pôle Emploi, PhD Student]
Michael Toth [Inria, PhD Student, from Dec 2019]

Interns and Apprentices

Arianna Corvi [Ipag business School de Nice, until Mar 2019]
Maxime Legoupil [ENS Ulm, from Jun 2019 until Jul 2019]
Clément Ogier [ENS Ulm, Jul 2019]
Adam Khayam [Inria, from Jan 2019 to Jun 2019]
Carlo Prato [Inria, from Apr 2019 to Sep 2019]

Administrative Assistant

Nathalie Bellesso [Inria, Administrative Assistant]

Visiting Scientists

Andrei Sabelfeld [Chalmers, Jul 2019]
Cristiana Santos [University of Toulouse, Feb 2019 and May 2019]

External Collaborator

Marc Feeley [Université de Montréal]

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, functional, and reactive 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.

As a *Hop* DSL, we have created *HipHop*, a synchronous orchestration language for web and IoT applications. *HipHop* facilitates the design and programming of complex web/IoT applications by smoothly integrating three computation models and programming styles that have been historically developed in different communities and for different purposes: *i) Transformational programs* that simply compute output values from input values, with comparatively simple interaction with their environment; *ii)* asynchronous concurrent programs that perform interactions between their components or with their environment with uncontrollable timing, using typically network-based communication; and *iii)* synchronous reactive programs that react to external events in a conceptually instantaneous and deterministic way.

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. Application Domains

4.1. Web

The Web is the natural application domain of the team. We are designing and implementing multitier languages for helping the development of Web applications. We are creating static and dynamic analyses for Web security. We are conducting empirical studies about privacy preservation on the Web.

4.2. Internet of Things

More recently, we have started focusing on *Internet of Things* (IoT) applications. They share many similarities with Web applications so most of the methodologies and expertises we have developed for the Web apply to IoT but the restricted hardware resources made available by many IoT devices demand new developments and new research explorations.

5. New Software and Platforms

5.1. 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/

5.2. 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

5.3. 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/

5.4. 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

5.5. 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: Francis Doliére Some
- URL: http://www-sop.inria.fr/members/Doliere.Some/essos/

5.6. webstats

Webstats

KEYWORDS: Web Usage Mining - Statistic analysis - Security

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: Francis Doliére Some
- URL: https://webstats.inria.fr

5.7. Skini

Platform for creation and execution for audience participative music

KEYWORDS: Music - Interaction - Web Application - Synchronous Language

FUNCTIONAL DESCRIPTION: Skini is a platform form designing et performing collaborative music. It is based on two musical concept: pattern and orchestration. The orchestration is design using HipHop.js.

RELEASE FUNCTIONAL DESCRIPTION: Can be use for performance and création.

- Contact: Bertrand Petit

5.8. Platforms

5.8.1. BehaviorTrack

Keyword: Web tracking detection, Large-scale measurement

Description: In our study, we propose a tracking detection method inspired by analyzing behavior of invisible pixels. By crawling 84,658 webpages from 8,744 domains, we detect that third-party invisible pixels are widely deployed: they are present on more than 94.51% of domains and constitute 35.66% of all third-party images. We propose a fine-grained behavioral classification of tracking based on the analysis of invisible pixels. BehaviorTrack uses this classification to detect new categories of tracking and uncover new collaborations between domains on the full dataset of 4,216,454 third-party requests.

- Contact: Imane Fouad
- URL: http://www-sop.inria.fr/members/Imane.Fouad/pixeltrack

6. New Results

6.1. JavaScript Implementation and Browser Security

We have pursued the development of *Hop* and our study on efficient and secure JavaScript implementations.

6.1.1. JavaScript Property Caches

JavaScript objects are dynamic. At any moment of their lifetime, properties can be added or deleted. In principle a property access requires a lookup in the object itself, and, possibly, in all the objects forming its prototype chain. All fast JavaScript implementations deploy strategies to implement this lookup operation in nearly constant time. They generally rely on two ingredients: *hidden classes* and *property caches*. Hidden classes describe object memory layouts. Property caches use these descriptions to access objects directly, avoiding the normal name lookup operations. Hidden classes and property caches make property accesses comparable in speed to field accesses of traditional languages like C and Java.

Hidden classes and property caches are not new. They were invented for Self, the first dynamically typed prototype-based languages, following Smalltalk's idea that already used caches at that time for optimizing method calls. For the past ten years they have enjoyed a revival of interest after it was shown how effective they are at improving Object-Oriented languages performance in general and specially JavaScript. Today most JavaScript implementations such as V8, JavaScriptCode, and SpiderMonkey use them. Hidden classes and property caches apply in specific situations, which unfortunately means that some accesses are unoptimized or not treated very efficiently.

1. **Property addition problem**: hidden classes support the accesses of existing properties but they do not handle efficiently property addition commonly found in object constructors.
2. **Prototype properties problem**: hidden classes and property caches optimize accesses of properties directly stored in the object. They do not optimize accesses of properties stored in one of the objects composing the prototype chain.
3. **Polymorphic properties problem**, as property caches require strict hidden class equivalence for optimizing accesses, polymorphic data structures and polymorphic method invocations need special treatment to not be left unoptimized. This has been addressed by the *Polymorphic Inline Cache* technique proposed by Holzle *et al.* in previous studies, which resorts to a dynamic search in the cache history. As a linear or binary search is involved, it is not as efficient as plain property caches.

Problem 1 is critical for all existing JavaScript programs as it impacts the performance of object construction. Problems 2 and 3 will become prominent with the advent of ECMAScript 6 class-like programming style that is backed up by object prototypes. We propose solutions to these problems. At the cost of one extra test inserted at each property access, we optimize prototype property accesses. Resorting to a static analysis, we propose a technique that we call *speculative caches* for optimizing object construction.

Trading memory space for speed, we propose *cache property tables* that enable accessing polymorphic objects in constant time. For the analogy with C++ virtual tables we call these cache tables *vtables*.

We have implemented these techniques in *Hopc*, the *Hop* static JavaScript compiler and we have presented them in a conference publication [17]. We have shown how the complement and enhance property caches used for accessing object properties of JavaScript like languages. We have shown that they take over classical caches when the searched property is either stored in an object of the prototype chain or defined using accessors. They also support efficiently polymorphic and megamorphic property accesses. Finally, they also support efficient object extensions. These techniques do not apply as frequently as simple property caches that cover a vast majority of accesses. However, since they impose no overhead when not used, they can be integrated in any existing system at no run time cost. We have validated the approach with an experimental report based that shown that the presented techniques improve performance in situations where simple cache miss.

6.1.2. Secure JavaScript

Whereas the dynamic nature of JavaScript plays an essential role in the advantages it offers for easy and fast development, a malicious JavaScript program can easily break the integrity and confidentiality of a web or IoT application. JavaScript dynamic semantics and sharing are deeply intricated and attacker code can trivially exploit these.

We have developed a compiler, called SecureJS to offer security guarantees for JavaScript on clients, servers, and IoT devices. Our compiler is applicable to ECMAScript 5th legacy code, which in particular means that we allow for built-in JavaScript functions. Moreover, we go beyond the JavaScript language and handle a common web API, XMLHttpRequest module. The challenge is to cover most of the JavaScript language efficiently while providing strong security guarantees. For the latter, we formally define and prove the compiler's security guarantees by means of a new security property, coined as *dynamic delimited release*, for JavaScript integrity and confidentiality.

Compiled programs can be effortlessly deployed in client, server, and IoT JavaScript environments and do not require an external isolation mechanism to preserve integrity and confidentiality.

We have validated SecureJS experimentally using ECMAScript Test262 test suits. First, we have shown that SecureJS preserves the correct SecureJS semantics. Second, we have shown that it successfully implements the memory isolation needed to enforce the security property.

The current SecureJS implementation as been architectured to support low-power platforms that only supports ECMAScript 5. In the future we plan to accommodate more recent version of JavaScript for the platforms that supports it. This will extend the possibility of communications between trusted and untrusted codes and this will enable more efficient implementation techniques. A paper describing this work is currently under submission.

6.1.3. Empowering Web Applications with Browser Extensions

Browser extensions are third party programs, tightly integrated to browsers, where they execute with elevated privileges in order to provide users with additional functionalities. Unlike web applications, extensions are not subject to the Same Origin Policy (SOP) and therefore can read and write user data on any web application. They also have access to sensitive user information including browsing history, bookmarks, credentials (cookies) and list of installed extensions. They have access to a permanent storage in which they can store data as long as they are installed in the user's browser. They can trigger the download of arbitrary files and save them on the user's device. For security reasons, browser extensions and web applications are executed in separate contexts. Nonetheless, in all major browsers, extensions and web applications can interact by exchanging messages. Through these communication channels, a web application can exploit extension privileged capabilities and thereby access and exfiltrate sensitive user information.

We have analyzed the communication interfaces exposed to web applications by Chrome, Firefox and Opera browser extensions [18]. As a result, we identified many extensions that web applications can exploit to access privileged capabilities. Through extensions' APIS, web applications can bypass SOP and access user data on any other web application, access user credentials (cookies), browsing history, bookmarks, list of installed extensions, extensions storage, and download and save arbitrary files in the user's device. Our results demonstrate that the communications between browser extensions and web applications pose serious security and privacy threats to browsers, web applications and more importantly to users. We discuss countermeasures and proposals, and believe that our study and in particular the tool we used to detect and exploit these threats, can be used as part of extensions review process by browser vendors to help them identify and fix the aforementioned problems in extensions.

6.2. Timing-side channels attacks

We have pursued our studies on foundations of language-based security following two axes on timing-side channels research:

6.2.1. Speculative constant time

The most robust way to deal with timing side-channels in software is via *constant-time* programming—the paradigm used to implement almost all modern cryptography. Constant-time programs can neither branch on secrets nor access memory based on secret data. These restrictions ensure that programs do not leak secret information via timing side channels, at least on hardware *without* microarchitectural features. However, microarchitectural features are a major source of timing side channels as the growing list of attacks (Spectre, Meltdown, etc) is showing. Moreover code deemed to be constant-time in the usual sense may in fact leak information on processors with microarchitectural features. Thus the decade-old constant-time recipes are no longer enough. We lay the foundations for constant-time in the presence of micro-architectural features that have been exploited in recent attacks: out-of-order and speculative execution. We focus on constant-time for two key reasons. First, *impact*: constant-time programming is largely used in narrow, high-assurance code—mostly cryptographic implementations—where developers already go to great lengths to eliminate leaks via side-channels. Second, *foundations:* constant-time programming is already rooted in foundations, with well-defined semantics. These semantics consider very powerful attackers have control over the cache and the scheduler. A nice effect of considering powerful attackers is that the semantics can already overlook many hardware details—e.g., since the cache is adversarially controlled there is no point in modeling it precisely—making constant-time amenable to automated verification and enforcement.

We have first defined a semantics for an abstract, three-stage (fetch, execute, and retire) machine. This machine supports out-of-order and speculative execution by modeling *reorder buffers* and *transient instructions*, respectively. Our semantics assumes that attackers have complete control over microarchitectural features (e.g., the branch target predictor), and uses adversarial execution *directives* to model adversary's control over predictors. We have then defined *speculative constant-time*, the counterpart of *constant-time* for machines with out-of-order and speculative execution. This definition has allowed us to discover microarchitectural side channels in a principled way—all four classes of Spectre attacks as classified by Canella et al., for example, manifest as violation of our constant-time property. Our semantics even revealed a new Spectre variant, that exploits the aliasing predictor. The variant can be disabled by unsetting a flag, by illusttrates the usefulness of our semantics. This study is described in a paper currently submitted.

6.2.2. Remote timing attacks

A common approach to deal with timing attacks is based on preventing secrets from affecting the execution time, thus achieving security with respect to a strong, *local* attacker who can measure the timing of program runs. Another approach is to allow branching on secrets but prohibit any subsequent attacker-visible side effects of the program. It is sometimes used to handle *internal timing* leaks, i.e., when the timing behavior of threads affects the interleaving of attacker-visible events via the scheduler.

While these approaches are compatible with strong attackers, they are highly restrictive for program runs as soon as they branch on a secret. It is commonly accepted that "adhering to constant-time programming is hard" and "doing so requires the use of low-level programming languages or compiler knowledge, and forces developers to deviate from conventional programming practices".

This restrictiveness stems from the fact that there are many ways to set up timing leaks in a program. For example, after branching on a secret the program might take different time in the branches because of: (i) more time-consuming operations in one of the branches, (ii) cache effects, when in one of the branches data or instructions are cached but not in the other branch, (iii) garbage collection (GC) when in one of the branches GC is triggered but not in the other branch, and (iv) just-in-time (JIT) compilation, when in one of the branches a JIT-compiled function is called but not in the other branch. Researchers have been painstakingly addressing these types of leaks, often by creating mechanisms that are specific to some of these types. Because of the intricacies of each type, addressing their combination poses a major challenge, which these approaches have largely yet to address.

This motivates a general mechanism to tackle timing leaks independently of their type. However, rather than combining enforcement for the different types of timing leaks for strong local attackers, is there a setting where the capabilities of attackers are perhaps not as strong, enabling us to design a general and less restrictive mechanism for a variety of timing attacks with respect to a weaker attacker?

We focus on timing leaks under *remote* execution. A key difference is that the remote attacker does not generally have a reference point of when a program run has started or finished, which significantly restricts attacker capabilities.

We illustrate remote timing attacks by two settings: a server-side setting of IoT apps where apps that manipulate private information run on a server and a client-side setting where e-voting code runs in a browser.

IFTTT (If This Then That), Zapier, and Microsoft Flow are popular IoT platforms driven by enduser programming. App makers publish their apps on these platforms. Upon installation apps manipulate sensitive information, connecting cyberphysical "things" (e.g., smart homes, cars, and fitness armbands) to online services (e.g., Google and Dropbox) and social networks (e.g., Facebook and Twitter). An important security goal is to prevent a malicious app from leaking private information of a user to the attacker.

Recent research identifies ways to leak private information by IoT apps and suggests tracking information flows in IoT apps to control these leaks. The suggested mechanisms perform data-flow (*explicit*) and control-flow (*implicit*) tracking. Unfortunately, they do not address timing leaks, implying that a malicious app maker can still exfiltrate private information, even if the app is subject to the security restrictions imposed by the proposed mechanisms.

In addition, Verificatum, an advanced client-side cryptographic library for e-voting motivates the question of remote timing leaks with respect to attackers who can observe the presence of encrypted messages on the network.

This leads us to the following general research questions:

1. What is the right model for remote timing attacks?
2. How do we rule out remote timing leaks without rejecting useful secure programs?
3. How do we generalize enforcement to multiple security levels?
4. How do we harden existing information flow tools to track remote timing leaks?
5. Are there case studies to give evidence for the feasibility of the approach?

To help answering these questions, we propose an extensional knowledge-based security characterization that captures the essence of remote timing attacks. In contrast to the local attacker that counts execution steps/time since the beginning of the execution, our model of the remote attacker is only allowed to observe inputs and outputs on attacker-visible channels, along with their timestamps. At the same time, the attacker is in charge of the potentially malicious code with capabilities to access the clock, in line with assumptions about remote execution on IoT app platforms and e-voting clients.

A timing leak is typically enabled by branching on a secret and taking different time or exhibiting different cache behavior in the branches. However, as discussed earlier, it is desirable to avoid restrictive options like forcing the execution to take constant time, prohibiting attacker-visible output any time after the branching, or prohibiting branching on a secret in the first place.

Our key observation is that for a remote attacker to successfully set up and exploit a timing leak, program behavior must follow the following pattern: (i) branching on a secret takes place in a program run, and either (ii-a) the branching is followed by more than one attacker-visible I/O event, or (ii-b) the branching is followed by one attacker-visible I/O event, and prior to the branching there is either an attacker-visible I/O event or a reading to the clock.

Based on this pattern, we design Clockwork, a monitor that rules out timing leaks. Our mechanism pushes for permissiveness. For example, runs (free of explicit and implicit flows) that do not access the clock and only have one attacker-visible I/O event are accepted.

Runs that do not perform attacker-visible I/O after branching on a secret are accepted as well. As we will see, these kinds of runs are frequently encountered in secure IoT and e-voting apps.

We implement our monitor for JavaScript, leveraging JSFlow, a state-of-the-art information flow tracker for JavaScript. We demonstrate the feasibility of the approach on a case study with IFTTT, showing how to prevent malicious app makers from exfiltrating users' private information via timing, and a case study with Verificatum, showing how to track remote timing attacks with respect to network attackers. Our case studies demonstrate both the security and permissiveness. While apps with timing leaks are rejected, benign apps that use clock and I/O operations in a non-trivial fashion are accepted.

6.3. Security analysis of ElGamal implementations

Throughout the last century, especially with the beginning of public key cryptography due to Diffie-Hellman, many cryptographic schemes have been proposed. Their security depends on mathematically complex problems such as integer factorization and discrete logarithm. In fact, it is thought that a cryptographic scheme is secure if it resists cryptographic attacks over a long period of time. On one hand, since certain schemes may take several years before being widely studied in depth, they become vulnerable as time passes. On the other hand, a cryptographic scheme is a provable one, if it resists cryptographic attacks relying on mathematical hypothesis.

Being easily adaptable to many kinds of cryptographic groups, the ElGamal encryption scheme enjoys homomorphic properties while remaining semantically secure , provided that the Decisional Diffie-Hellman (DDH) assumption holds on the chosen group. While the homomorphic property forbids resistance against chosen ciphertext attacks, it is very convenient for voting systems. The ElGamal encryption scheme is the most extensively used alternative to RSA, and it is the homomorphic encryption scheme almost exclusively used for voting systems. Moreover, ElGamal is the only homomorphic encryption scheme implemented by default in many hardware security modules.

In order to be provable secure, ElGamal encryption needs to be implemented on top of a group verifying the Decisional Diffie-Hellman (DDH) assumption. Since this assumption does not hold for all groups, one may have to wrap an encoding and a decoding phase to ElGamal to be able to have a generic encryption scheme.

We have submitted a paper that studies ElGamal encryption scheme libraries in order to identify which implementations respect the DDH assumption. The paper presents an analysis of 25 libraries that implement ElGamal encryption scheme in the wild. We focus our analysis on understanding whether the DDH assumption is respected in these implementations, ensuring a secure scheme in which no information about the original message could be leaked. The DDH assumption is crucial for the security of ElGamal because it ensures indistinguishability under chosen-plaintext attacks (IND-CPA). Without the DDH assumption, encryption mechanisms may leak one bit of information about the plaintext and endager the security of the electoral system as one bit has the ability to completely invalidate privacy in an election. One way to comply with the DDH assumption is by using groups of prime order. In particular, when adopting safe primes, one can ensure the existence of a *large* prime order subgroup and restrict messages to belong to this subgroup. Mapping plaintexts into subgroups is called message encoding. Such encoding necessitates to be efficient and precisely invertible to allow decoding after the decryption.

Our results show that out of 25 analyzed libraries, 20 are wrongly implemented because they do not respect the conditions to achieve IND-CPA security under the DDH assumption. This means that encryptions using ElGamal from any of these 20 libraries leak one bit of information.

From the 5 libraries which respect the DDH assumption, we also study and compare various encoding and decoding techniques. We identify four different message encoding and decoding techniques and discuss the different designs and conclude which implementation is more efficient for voting systems.

6.4. Measurement and Detection of Web Tracking

6.4.1. Missed by Filter Lists: Detecting Unknown Third-Party Trackers with 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 [1] and EasyPrivacy [2] (EL&EP) are the most popular publicly maintained blacklist of know advertising and tracking domains, used by the popular browser extensions AdBlock Plus [3] and uBlockOrigin [4]. Disconnect [5] is another very popular list for detecting domains known for tracking, used in Disconnect browser extension [6] and in integrated tracking protection of Firefox browser. 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.

Our contributions: To evaluate the effectiveness of filter lists, we propose a new, fine-grained behavior-based tracking detection. Our results are based on a stateful dataset of 8K domains with a total of 800K pages generating 4M third-party requests. We make the following contributions:

- *We analyse all the requests and responses that lead to invisible pixels (by "invisible pixels" we mean* 1×1 *pixel images or images without content).* Pixels are routinely used by trackers 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 tag into a webpage. This makes invisible pixels *the perfect suspects for tracking* and propose a new classification of tracking behaviors. Our results show that pixels are still widely deployed: they are present on more than 94% of domains and constitute 35.66% of all third-party images. We found out that pixels are responsible only for 23.34% of tracking requests, and the most popular tracking content are scripts: a mere loading of scripts is responsible for 34.36% of tracking requests.
- *We uncover hidden collaborations between third parties.* We applied our classification on more than 4M third-party requests collected in our crawl. We have detected new categories of tracking and collaborations between domains. We show that domains sync first party cookies through a *first to third party cookie syncing*. This tracking appears on 67.96% of websites.
- *We show that filter lists miss a significant number of cookie-based tracking.* Our evaluation of the effectiveness of EasyList&EasyPrivacy and Disconnect lists shows that they respectively miss 25.22% and 30.34% of the trackers that we detect. Moreover, we find that if we combine all three lists, 379,245 requests originating from 8,744 domains still track users on 68.70% of websites.
- *We show that privacy browser extensions miss a significant number of cookie-based tracking.* By evaluating the popular privacy protection extensions: Adblock, Ghostery, Disconnect, and Privacy Badger, we show that Ghostery is the most efficient among them and that all extensions fail to block at least 24% of tracking requests.

[1] https://easylist.to/
[2] https://easylist.to/easylist/easyprivacy.txt
[3] https://adblockplus.org/
[4] https://github.com/gorhill/uBlock
[5] https://disconnect.me/trackerprotection/blocked
[6] https://disconnect.me/

This paper [15] has been accepted for publication at the Privacy Enhancing Technologies Symposium (PETs) 2020.

6.4.2. 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.

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.

6.5. Security Analysis of GDPR Subject Access Request Procedures

With the GDPR in place since May 2018, the rights of the European users have been strengthened. The GDPR defines users' rights and aims at protecting their personal data. Every European Data Protection Authority (DPA) provides advices, explanations and recommendations on the use of these rights. However, the GDPR does not provide any prescriptive requirements on how to authenticate a data subject request. This lack of concrete description undermines the practical effect of the GDPR: it hampers the way to exercise the subject access right, to check the lawfulness of the processing and to enforce the derived legal rights therefrom (erasure, rectification, restriction, etc).

Every data subject would like to benefit from the rights specified in GDPR, but still wonders: *How do I exercise my access right? How do I prove my identity to the controller?* These questions are critical to build trust between the data subject and the controller. The data subject is concerned with threats like *impersonation* and *abusive identity check*. Impersonation is the case of a malicious party who attempts to abuse the subject access request (SAR) by impersonating a subject to a controller. Abusive identity check occurs when a data controller is too curious and verifies the identity of a subject by asking irrelevant and unnecessary information like an electricity bill or government issued documents.

Symmetrically, every data controller needs to know how to proceed when they receive an access request: *Is the request legitimate? What is necessary to identify the subject's data?* These concerns aggravate when controllers deal with indirectly-linked identifiers, such as IP addresses, or when they have no prior contact with data subjects, as in *Google Spain* [7]. Most of all, data controllers want to avoid data breaches, as it can result in legal proceedings and heavy fines. Such consequence occurs in two cases: *(i)* the data controller releases data to an illegitimate subject, or *(ii)* he releases data of a subject A to a legitimate subject B.

All these questions concern the authentication procedure between the data subject and the controller. They both share a common interest in holding a strong authentication procedure to prevent impersonation and data breaches. The subject must be careful during the authentication procedure, as for providing too much personal information could compromise her right of privacy. Additionally, the controller needs to ask the appropriate information to identify the subject's data without ambiguity. There is clearly a tension during this authentication act between the controller, who tries to get as much information as possible, and the data subject who wants to provide as little as possible. Plausibly, subject access rights can probably increase the incidence of personal records being accidentally or deliberately opened to unauthorised third parties [22].

This work studies *the tension during the authentication between the data subject and the data controller*. We first evaluate the threats to the SAR authentication procedure and then we analyze the recommendations of 28 DPAs of European Union countries. We observe that four of them can potentially lead to abusive identity check. On the positive side, six of them are recommending to enforce the data minimization principle during authentication. This principle, on one hand, protects the right to privacy of data subjects, and on the other hand prevents data controllers to massively collect personal data that is not needed for authentication, thus preventing abusive identity check.

We have then evaluated the authentication procedure when exercising the access right of the 50 most popular websites and 30 third-party tracking services. Several popular websites require to systematically provide a national identity card or government-issued documents to authenticate the data subject. Among third-party tracking services, 9 of them additionally to cookies demand other personal data from the data subjects, like the identity card or the full name. We explain that such demands are not justified because additional information can not prove the ownership of the cookie.

We then provide guidelines to Data Protection Authorities, website owners and third party services on how to authenticate data subjects safely while protecting their identities, and without requesting additional unnecessary information (complying with the data minimization principle). More precisely, we explain how data controllers and data subjects must interact and how digital identifiers can be redesigned to be compliant with the GDPR.

This work has been published at the Annual Privacy Forum (APF) 2019 [13].

6.6. Measuring Legal Compliance of Cookie Banners

6.6.1. Deciphering EU legal requirements on consent and technical means to verify compliance of cookie banners

In this work, we analyze the legal requirements on how cookie banners are supposed to be implemented to be fully compliant with the ePrivacy Directive and the GDPR.

Our contribution resides in the definition of 17 operational and fine-grained requirements on cookie banner design that are legally compliant, and moreover, we define whether and when the verification of compliance of each requirement is technically feasible.

[7] Google Spain SL and Google Inc. v Agencia Española de Protección de Datos (AEPD) and Mario Costeja González, Case C-131/12, https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:62012CJ0131&from=EN

The definition of requirements emerges from a joint interdisciplinary analysis composed of lawyers and computer scientists in the domain of web tracking technologies. As such, while some requirements are provided by explicitly codified legal sources, others result from the domain-expertise of computer scientists. In our work, we match each requirement against existing cookie banners design of websites. For each requirement, we exemplify with compliant and non-compliant cookie banners.

As an outcome of a technical assessment, we verify per requirement if technical (with computer science tools) or manual (with any human operator) verification is needed to assess compliance of consent and we also show which requirements are impossible to verify with certainty in the current architecture of the Web. For example, we explain how the GDPR's requirement for revocable consent could be implemented in practice: when consent is revoked, the publisher should delete the consent cookie and communicate the withdrawal to all third parties who have previously received consent.

With this approach we aim to support practically-minded parties (compliance officers, regulators, privacy NGOs, researchers, and computer scientists) to assess compliance and detect violations in cookie banners' design and implementation, specially under the current revision of the EU ePrivacy framework.

This working paper is submitted for publication.

6.6.2. Measuring Legal Compliance of Banners from IAB Europe's Transparency and Consent Framework

As a result of the GDPR and the ePrivacy Directive, (known as “cookie law”), European users encounter cookie banners on almost every website. Many of such banners are implemented by Consent Management Providers (CMPs), who respect the IAB Europe's Transparency and Consent Framework (TCF). Via cookie banners, CMPs collect and disseminate user consent to third parties. In this work, we systematically study IAB Europe's TCF and analyze consent stored behind the user interface of TCF cookie banners. We analyze the GDPR and the ePrivacy Directive to identify legal violations in implementations of cookie banners based on the storage of consent and detect such violations by crawling 22 949 European websites.

With two automatic and semi-automatic crawl campaigns, we detect violations, and we find that: 175 websites register positive consent even if the user has not made their choice; 236 websites nudge the users towards accepting consent by pre-selecting options; and 39 websites store a positive consent even if the user has explicitly opted out. Performing extensive tests on 560 websites, we find at least one violation in 54% of them.

Finally, we provide a browser extension called "Cookie Glasses" to facilitate manual detection of violations for regular users and Data Protection Authorities.

This working paper is submitted for publication at an international conference.

6.7. Session Types

Session types describe communication protocols between two or more parties by specifying the sequence of exchanged messages and 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 been subsequently embedded into a range of functional, concurrent, and object-oriented programming languages.

While binary sessions can be described by a single session type, multiparty sessions require two kinds of types: a *global type* that describes the whole session protocol, and *local types* that describe the contributions of the various participants to the protocol. The key requirement to achieve safety properties such as the absence of communication errors and deadlock-freedom, is that the local types of the processes implementing the participants be obtained as projections from the same global type (the one describing the session protocol).

We have pursued our work on multiparty session types along four main directions, in collaboration with colleagues from the Universities of Groningen, Luxemburg, Nice Sophia Antipolis, Turin and Eastern Piedmont. One of these directions is described in Section 6.8.3, the others are described below.

6.7.1. 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 may 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 Milner's calculus CCS, then for the π-calculus, and only recently for typed session calculi.

Following up on our previous work on concurrent reversible sessions, 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 a choice. This operator was inspired by the notion of *connecting communication* introduced by other authors to describe protocols with optional participants. Our calculus 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, where only some specific participants, the "choice leaders", can trigger a rollback. We present a session type system for this calculus and show that it enforces the expected properties of session fidelity, forward progress and backward progress. This work has been published in the journal [11].

6.7.2. Multiparty Sessions with Internal Delegation

We have investigated a new form of *delegation* for multiparty session calculi. Usually, the delegation mechanism 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 multiparty sessions. As a consequence, properties such as deadlock-freedom or lock-freedom are difficult to ensure in the presence of delegation. In our work, 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 (whence the name "internal delegation") and may be captured by its global type. To increase flexibility in the use of delegation, we use again connecting communications, in order to accommodate optional participants in the branches of choices. By this means, we are also 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 has been published in a special issue of TCS dedicated to Maurice Nivat [12].

6.7.3. Event Structure Semantics for Multiparty Sessions

In the work [14] we investigate the relationship between multiparty session calculi and other concurrency models, by focussing on Event Structures as proposed in the late 80's. We consider a standard multiparty session calculus where sessions are described as networks of sequential processes, and each process implements a participant in the session. We propose an interpretation of such networks as *Flow Event Structures* (FESs) (a subclass of Winskel's Stable Event Structures), which allows concurrency between session communications to be explicitly represented. We then introduce global types for these networks, and define an interpretation of global types as *Prime Event Structures* (PESs). Since the syntax of global types does not allow all the concurrency among communications to be expressed, the events of the associated PES need to be defined as equivalence classes of communication sequences up to *permutation equivalence*. We show that when a network is typable by a global type, the FES semantics of the former is equivalent, in a precise technical sense, to the PES semantics of the latter.

This work has been published in a volume dedicated to Rocco De Nicola on the occasion of his 65th birthday [14]. An extended version is available as Research Report [21].

6.8. Web Reactive Programming

6.8.1. HipHop.js

This year, we have completed the design of the *HipHop* programming language. We have finalized the syntax of core instructions, stabilized the interfacing with JavaScript, added variables that supplement signals in local computation, and we have completed the synchronous/asynchronous connections. A paper describing this final version of the paper is currently under submission.

We have also improved significantly the *HipHop* implementation for speed and for debugging.

- Leveraging on the *Hop* speed improvement and by adding a new *HipHop* compilation stage we have been able to accelerate by a factor of about 10× the intrinsic execution time of the reactive machine. The optimization removes nets of the virtual electronic circuits that are generated by the *HipHop* compiler by propagating constant and by collapsing identical nodes. This contributions is included in the main development tree (https://github.com/manuel-serrano/hiphop).
- A central difficulty of the synchronous reactive programming is debugging and error messages. The *HipHop* compilation roughly consists in implementing efficiently and compactly a deterministic automata that represents the user source code. If a causality error is detected during that compilation, unless a precise isolation of the user source code fragments that are involved in that error, the error message reported to the user is so imprecise that fixing the problem is difficult. We have implemented an algorithm based on *strongly connected components* that enables the needed isolation. This experimental feature is currently publicly available via a dedicating development branch under the *HipHop* github repository.

6.8.2. Interactive music composition

The production of a piece of music by school children using the Skini platform as part of SACEM's call for projects "Fabrique à musique" (Music Factory) was initiated in 2019. It ended in 2019 with the realization of a show at the Nice Conservatory in May 2019. The music piece thus created implemented all of Skini's functionalities, from the distributed sequencer that allowed the pupils to design the basic material, to the control of the live orchestration, not by the audience in this case, but by the 24 students who participated in the project. Following the success of this first experiment, the project is being extended for 2020/2021 by Inria, with another class, as part of the "les cordées de la réussite" program.

Beyond the improvement of the system, and in particular of the distributed sequencer, thanks to the significant performance improvements of HipHop.js, it has been possible to enrich the controls on musical orchestrations by driving transformation elements of Skini's basic elements (the patterns) such as transpositions, use of patterns of different durations, music mode conversions, or tempo control. The coupling of these new processes has enriched the range of possibilities for interaction and has opened up new horizons in the field of pattern-based generative music.

In terms of musical creation, we have been able to implement orchestrations using the platform's new technical possibilities in order to reduce musical processes perceived as too automatic. We can note as convincing results: the possibility to break the too big symmetries on the durations of the patterns, and the variations of tempi subjected to various controls. We were able to demonstrate that with the same HipHop.js music orchestration program, we could efficiently generate very different musical pieces.

Skini music composition has been described in a conference paper presented in the NIME 2019 conference [16].

6.8.3. Multiparty Reactive Sessions

Ensuring that communication-centric systems interact according to an intended protocol is a challenging 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 timed reactive systems. *Session-based Concurrency* is the model of concurrent computation induced by session types.

In the Research Report [20], we propose a multiparty session calculus enriched with features from SRP. In this calculus, protocol participants may broadcast messages, suspend themselves while waiting for a message, and react to events. Our main contribution is a session type system for this calculus, which enforces session correctness for non-interleaved sessions and additionally ensures *input timeliness*, a time-related property that entails livelock-freedom (while deadlock-freedom holds by construction in our calculus). Our type system departs significantly from existing ones, specifically as it captures the notion of “logical instant” typical of SRP.

7. Bilateral Contracts and Grants with Industry

7.1. 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. Nataliia Bielova, Manuel Serrano and Tamara Rezk are involved in this project. 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. Yoon Seok Ko has worked on this project as a postdoc.

8. Partnerships and Cooperations

8.1. Regional Initiatives

8.1.1. Skini

Skini was used for the production of a musical piece as part of SACEM’s “Music Factory” program in collaboration with the *CIRM* in Nice and the *Conservatory of Nice*. This piece was designed, and produced in May at the Nice Conservatory, by 12 years old pupils of the Nucéra secondary school in Nice after a dozen working sessions within the school. This production is followed by a similar project with 14 years old pupils as part of the "Cordé de la résussite" programme run by Inria with the objective of a musical production in spring 2020.

8.2. National Initiatives

8.2.1. 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.

8.2.2. ANR PrivaWeb

The PrivaWeb project (Privacy Protection and ePrivacy Compliance for Web Users) is funded by the ANR JCJC program for 48 months, started 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.

8.2.3. PIA ANSWER

The ANSWER project (Advanced aNd Secured Web Experience and seaRch) is funded by PIA program for 36 months, starting January 1, 2018. The aim of the ANSWER project is to develop the new version of the http://www.qwant.com search engine by introducing radical innovations in terms of search criteria as well as indexed content and users' privacy. The partners of this project include QWANT and Inria teams Wimmics, Indes, Neo and Diana.

8.3. Inria Internal Funding

8.3.1. 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.

8.3.2. AEx DATA4US

DATA4US is a joint project between two teams in Inria Sophia Antipolis and Inria Grenoble - Rhône-Alpes that tackles these interdisciplinary challenges by establishing collaborations with researchers in Law. Members are Nataliia Bielova (INDES) and Cedric Lauradoux (Privatics).

DATA4US will propose a new architecture for exercising access rights that will explain the users whether their data has been legally collected and eventually help contact DPAs for further investigations.

8.3.3. 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 is part of this project.

8.4. European Initiatives

8.4.1. H2020 Sparta

SPARTA (Strategic Programs for Advanced Research and Technology in Europe) is a novel cybersecurity competence network, with the objective to collaboratively develop and implement top-tier research and innovation actions. Strongly guided by concrete challenges forming an ambitious Cybersecurity Research & Innovation Roadmap, SPARTA will tackle hard innovation challenges, leading the way in building transformative capabilities and forming a world-leading cybersecurity competence network across the EU. Four initial research and innovation programs will push the boundaries to deliver advanced solutions to cover emerging issues, with applications from basic human needs to economic activities, technologies, and sovereignty.

See also: https://www.sparta.eu/

8.4.2. Collaborations in European Programs, Except FP7 & H2020

8.4.2.1. 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 was the first European network of excellence aimed at coordinating research on reversible computation.

See also: http://www.revcomp.eu

8.4.2.2. 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 was 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 objective of the project was to study formal models for reliable distributed communication-centric software systems. The project focussed on analysis and validation techniques based on behavioural types, aimed at enforcing various properties (safety, liveness, security) of structured communications.

8.5. International Initiatives

8.5.1. Inria International Partners

8.5.1.1. Informal International Partners

- We are collaborating 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. Benoit Baudry from KTH Royal Institute of Technology, Sweden and with Pierre Laperdrix from Stony Brook University on the survey of browser fingerprinting technologies.
- We are setting a new collaboration with Dr. Zinaida Benenson from University of Erlangen-Nuremberg, Germany, to study Human Factors in Privacy: in particular, to set up user studies to evaluate their perception and understanding of the cookie banners design and measure the influence of dark patterns on user decisions.

- We are setting a new collaboration with Prof. Martin Johns from TU Braunschweig, Germany, to work on cryptographic primitives to include proof of ownership in browser cookies that would facilitate the exercise of GDRP subject access rights. The is a joint collaboration with Cedric Lauradoux from Privatics.
- 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. This year, this collaboration was extended to Dr. Ross Horne from the University of Luxemburg. 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 pursuing our collaboration on reactive programming and on higher contracts for security with Prof. Robby Findler from Northwestern University in Chicago.
- We are pursuing our collaboration with Prof. Marc Feeley from Univerisity of Montréal on the compilation of dynamic languages.

8.5.2. Participation in Other International Programs

8.5.2.1. International Initiatives

DAJA

Title: Detection strategies based on Software Metrics for Multitier JavaScript

International Partners (Institution - Laboratory - Researcher):

Universidad de Chile (Chile), Intelligent Software Construction laboratory (ISCLab) - Alexandre Bergel

Universidad Nacional del Centro de la Provincia de Buenos Aires (Argentina) Computer Science Departement - 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.

8.6. International Research Visitors

8.6.1. Visits of International Scientists

- We are collaborating with Cristiana Teixeira Santos from University Toulouse 1-Capitole. Cristiana is a postdoc in Data Protection Law with whom we have been analyzing legal requirements for GDPR consent, and cookie banners in particular. Cristiana has visited us two times in 2019 and will be hired as a postdoc for Inria AEx project DATA4US in 2020.
- As part of our ongoing collaboration on GDPR Subject Access Rights, Cedric Lauradoux has visited us several times in 2019, to expand our existing work [13] and establish new research directions. Cedris is a co-PI for Inria AEx project DATA4US.

- We are collaborating with Prof. Marc Feeley from University of Montréal. For the third consecutive year, M. Feeley has visited us for studying implementation of dynamic languages, and in particular we started a study of the efficient compilation of the Python programming language.
- We are collaborating with Prof. Andrei Sabelfeld from Chalmers University of Technology. A.Sabelfeld has visited us for one month in July 2019 for studying remote timing attackers in the context of IoT frameworks.
- Prof. Robby Findler and his PhD student Spencer Florence visited us in July, where we have organized a mini-workshop during a week, working with Prof. G. Berry and J. Krishnamurthy on the semantics and implementation of reactive languages.

8.6.1.1. Internships

- Nataliia Bielova has co-supervised Hicham Lesfari for 3 months together with Frederic Giroire from Inria Coati team.
- Nataliia Bielova has supervised the intern Michael Toth as a "relai-de-these" for 2 months.
- Ilaria Castellani and Tamara Rezk supervised the intern Carlo Prato for 6 months.
- Tamara Rezk supervised the ENS L3 internship of Maxime Legoupil for 7 weeks in June and July 2019.
- Tamara Rezk has co-supervised the ENS L3 internship of Clément Ogier in July 2019.
- Tamara Rezk supervised the internship of Adam Khayam for 6 months.
- Tamara Rezk supevised -as tutor- the internship of Ayoub Ider Aghbal in a company.

8.6.2. Visits to International Teams

For the third consecutive year, Manuel Serrano and Gérard Berry visited Prof. Robby Findler at University of Northeastern in Chicago. This time, Tamara Rezk joined the delegation that also visited Prof. Christos Dimoulas also working at Northeastern University. The Indes team and Findler's team have applied for the second time to the Inria Associated Team program.

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Scientific Events: Organisation

9.1.1.1. General Chair, Scientific Chair

- Ilaria Castellani was the co-chair (together with Mohammad Reza Mousavi, Antonio Ravara and Alexandra Silva) of the workshop “Open Problems in Concurrency Theory 2019” (OPCT 2019), which was held in affiliation with the POPL 2019 conference in Lisbon, on January 14-15, 2019. https://popl19.sigplan.org/track/opct-2019-papers
- Ilaria Castellani was the co-chair (together with Mohammad Reza Mousavi) of the workshop TRENDS 2019, the annual event of the IFIP WG1.8 on Concurrency Theory, which took place in Amsterdam on August 31, 2019, in association with the CONCUR 2019 conference. https://concurrency-theory.org/events/workshops/trends
- Nataliia Bielova was the co-chair (together with François Pellegrini) of the CNIL-Inria Privacy Award 2019. The award ceremony will take place at the CPDP 2020 conference in Brussels. https://www.cnil.fr/en/launch-4th-edition-cnil-inria-privacy-award

9.1.1.2. Member of the Organizing Committees

Manuel Serrano organized the IFIP 2.16 (working group on Language Design) workshop in Nice from November 11th to November 15th http://program-transformation.org/WGLD/NiceMeeting2019.

9.1.2. Scientific Events: Selection

9.1.2.1. Member of the Conference Program Committees

- Ilaria Castellani served in the Program Committees of the conference CONCUR 2019 and the workshop EXPRESS/SOS 2019.
- Nataliia Bielova served in the Program Committees of the workshop MADWeb 2019, and the conferences PETs 2019, IEEE SecDev 2019, and IEEE EuroS&P 2019.
- Manuel Serrano served in the Program Committee of ProWeb'19 workshop. He was member of the ACM Software award https://www.sigplan.org/Awards/Software/.
- Tamara Rezk served in the Program Committees of the workshops MADWeb 2019, SSIoT 2019, PriSC 2019, and the conferences IEEE SecDev 2019, and NDSS 2019, and Programming.

9.1.2.2. Reviewer

- The team members have been reviewers for the following conferences and workshops: CONCUR'19, EXPRESS/SOS'19, PETs 2019, IEEE SecDev 2019, IEEE EuroS&P 2019, OOPSLA'19, ...

9.1.3. Journal

9.1.3.1. Member of the Editorial Boards

- Ilaria Castellani was a member of the editorial board of *Technique et Science Informatiques*, a French journal ended in June 2019.
- Ilaria Castellani (together with Mohammad Reza Mousavi) was guest editor for the JLAMP special issue on Trends in Concurrency Theory (Selected invited contributions from the workshops TRENDS 2015 and TRENDS 2016), J. Log. Algebr. Meth. Program., vol. 107, 2019 [19]. https://doi.org/10.1016/j.jlamp.2019.07.001.

9.1.3.2. Reviewer - Reviewing Activities

- The team members have been reviewers for the following journal: The Computer Journal.

9.1.4. Invited Talks

- Nataliia Bielova has been a keynote at the the Francophone workshop on Privacy Protection " l'Atelier sur la Protection de la Vie Privée" (APVP), which took place in Cap Hornu (France) from 9 to 11 July 2019. https://project.inria.fr/apvp2019/. Nataliia gave invited talks at SAP Labs (France), NOYB and TU Wien (Austria) in July/August 2019. She was an invited speaker at the Mozilla Security Summit on November 8, 2019, Vienna (Austria). https://events.mozilla.org/mozillasecurityresearchsummit2019 presenting the joint work with Imane Fouad and Arnaud Legout.
- Celestin Matte presented an ongoing work at the CNIL on October 1, 2019. Title of the talk: "Détection des violations du RGPD et de la directive e-Privacy dans les bannières de consentement aux cookies du Transparency and Consent Framework d'IAB Europe".
- Manuel Serrano has been a invited to be a keynote at the *Huawei European Research Symposium* in January 2019. He presented his work on *Hop* and *HipHop*. He has been invited by RainCode, a Belgium company located in Brussels, specialized in language compilation, to give a talk on the static compilation of dynamic languages.

9.1.5. Leadership within the Scientific Community

- Ilaria Castellani is the chair of the IFIP TC1 WG 1.8 on Concurrency Theory since June 2014 (reelected for a second term in March 2018).
- Ilaria Castellani was a member of the Management Committee of the COST Action IC1405 on Reversible Computation (until April 2019).
- Nataliia Bielova is a member of the Steering Committee of ACM PLAS.

- Tamara Rezk was a member of the Steering Committee of the POST conference during 2019 and is a member of the Steering Committee of the PriSC workshop.

9.1.6. Research Administration

- 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 "Comité Scientifique du Colloquium". Within UCA, she is a member of the "Réseau Égalité" and of the organising committee of the seminar series "Forum Numerica".
- Ilaria Castellani was the chair of the "jury d'admissibilité" of the CRCN competition (for junior researcher permanent positions) in the Inria Centre of Grenoble Rhône Alpes.
- Nataliia Bielova was a member of the hiring committee for an Inria chair position "enseignement et recherche dans le domaine de la cyberse´curité" of Supelec in February 2019.
- Nataliia Bielova is a member of "Comité du Suivi Doctoral (CSD)" (Supervision of PhD students) of the Inria Sophia Antipolis Mediterranée research center.
- Since summer 2019, Manuel Serrano is vice-chair of the Inria evaluation committee (EC). The charge for this duty amounts to about 50% of his professional activities. He co-chaired the three research promotion juries of 2019. He chaired the Inria *Algorithmics, Computer Algebra and Cryptology* theme EC evaluation. He prepared the evaluation seminar of the Inria theme *Embedded systems, Architecture, and Compilation*.
- Tamara Rezk was a member of " Commission de Développement Technologique (CDT)" of the Inria Sophia Antipolis Mediterranée research center.

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

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

IUT : Héloïse Maurel, Introduction aux technologies du Web, 20ETD, niveau DUT1, University of Nice Sophia Antipolis, France

DIU-EIL : Héloïse Maurel, Introduction aux langages de description et programmation du Web , 20ETD, University of Nice Sophia Antipolis, France

Master: Nataliia Bielova, Privacy on the Internet, 15ETD, M2, SKEMA Business school, France.

Master : Nataliia Bielova, Security and Ethical aspect of Data, 27ETD, niveau M1, Université Côte d'Azur, France

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

9.2.2. E-learning

E-learning

Mooc: Nataliia Bielova with C. Lauradoux and V. Roca, 1 session (2 months), FUN-MOOC, Inria, public targeted: around 30,000 for all the sessions since 2018, https://www.fun-mooc.fr/courses/course-v1:inria+41015+session03/about.

9.2.3. Supervision

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

PhD in progress: Michael Toth, Privacy Policies and Cookie Banners as a GDPR consent, 1/12/2019, Nataliia Bielova and Vincent Roca.

Postdoc, Celestin Matte, Large-scale measurement of Cookie Banners, 1/03/2019-, Nataliia Bielova.

Postdoc: Yoon Seok Ko, Subsets of secure JavaScript, 1/10/2018-, Tamara Rezk, Manuel Serrano.

PhD in progress: Jayanth Krishnamurthy, Secure Reactive Web Programming, 12/09/2018, Manuel Serrano.

PhD in progress: Bertrand Petit, Musique Massivement Interactive, 12/09/2017, 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

PhD in progress : Adam Khayam, Semantics of Multitier Languages, 1/07/2019, Alan Schmitt and Tamara Rezk

Postdoc: Francis Somé, IoT secure broadcasting, 1/11/2018-1/04/2019, Tamara Rezk

9.2.4. Juries

- Nataliia Bielova was a member of the PhD jury of Antoine Vastel, University of Lille.
- Nataliia Bielova was a member of the jury "soutenance de stage" of the UBINET master of the University of Nice Sophia Antipolis.
- Tamara Rezk was a member of the HdR jury of Catalin Hritcu, ENS.
- Tamara Rezk was a reporter and jury member of the PhD jury of Alexandre Dang, University of Bretagne Loire.
- Tamara Rezk was a member of the jury “soutenance de stage” of the CASPAR master of the University of Nice Sophia Antipolis.
- Manuel Serrano was a reporter and jury member of N. Oostvogels’s PhD, Vrije University, Brussels.

9.3. Popularization

9.3.1. Internal or external Inria responsibilities

Tamara Rezk was member of the editorial board of Interstices (https://interstices.info/) and Blog Binaire Le Monde (https://www.lemonde.fr/blog/binaire/).

9.3.2. Articles and contents

- Nataliia Bielova has been interviewed by the Usine Digitale on 22 November 2019, title "Fuite massive de données à AccorHotels : ce que risque l’entreprise", https://www.usine-digitale.fr/article/fuite-massive-de-donnees-a-accorhotels-que-risque-l-entreprise.N906279.
- Nataliia Bielova has been interviewed by the L’OBS on 29 November 2019, title "Entre réalité augmentée et reconnaissance faciale, on a parcouru le supermarché de demain", https://www.nouvelobs.com/economie/20191129.OBS21725/entre-realite-augmentee-et-reconnaissance-faciale-on-a-parcouru-le-supermarche-de-demain.html.
- Nataliia Bielova and Tamara Rezk have contributed to the Inria white book on CyberSecurity, published in January 2019: https://hal.inria.fr/hal-01993308.
- The browser extension "Cookie Glasses" developed by Celestin Matte and Nataliia Bielova has been extensively discussed on the general public media due to a complaint made by NOYB to the CNIL on the GDPR and ePrivacy violations of several e-commerce websites: https://noyb.eu/say-no-to-cookies-yet-see-your-privacy-crumble/.

9.3.3. Education

- Nataliia Bielova: *Moderation and Animation of the MOOC Protection de la vie privée dans le monde numérique*, 05/2019-06/2019.

9.3.4. Interventions

- Nataliia Bielova made an intervention with high school students during Semaine De La Science with Cinesciences at the cinémathèque of Nice on 4 October 2019.
- Celestin Matte presented his work on "Tracage Wifi et Bluetooth" at the public event called "Pas Sage en Seine" in June 2019, https://programme.passageenseine.fr/.
- Nataliia Bielova made a presentation with BSc and MSc students during the ACM-W Ukrainian Chapter celebration on December 5, 2019, https://women.acm.org/2019-2020-celebrations/.
- Imane Fouad presented her work on "detection of third-party trackers" at Inria PhD Seminars in December 2019.

10. 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$^{\text{o}}$ 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$^{\text{o}}$ 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$^{\text{o}}$ 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$^{\text{o}}$ 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

Articles in International Peer-Reviewed Journals

[11] I. CASTELLANI, M. DEZANI-CIANCAGLINI, P. GIANNINI. *Reversible sessions with flexible choices*, in "Acta Informatica", November 2019, vol. 56, n^{o} 7-8, pp. 553-583 [*DOI :* 10.1007/S00236-019-00332-Y], https://hal.inria.fr/hal-02420508

[12] I. CASTELLANI, M. DEZANI-CIANCAGLINI, P. GIANNINI, R. HORNE. *Global types with internal delegation*, in "Theoretical Computer Science", September 2019 [*DOI :* 10.1016/J.TCS.2019.09.027], https://hal.inria.fr/hal-02419937

International Conferences with Proceedings

[13] C. BONIFACE, I. FOUAD, N. BIELOVA, C. LAURADOUX, C. SANTOS. *Security Analysis of Subject Access Request Procedures How to authenticate data subjects safely when they request for their data*, in "APF 2019 - Annual Privacy Forum", Rome, Italy, June 2019, pp. 1-20, https://hal.inria.fr/hal-02072302

[14] I. CASTELLANI, M. DEZANI-CIANCAGLINI, P. GIANNINI. *Event Structure Semantics for Multiparty Sessions*, in "Models, Languages, and Tools for Concurrent and Distributed Programming - Hommage to Rocco De Nicola on the Occasion of His 65th Birthday", Lucca, Italy, Lecture Notes in Computer Science, Michele Boreale and Flavio Corradini and Michele Loreti and Rosario Pugliese, July 2019, vol. 11665, pp. 340-363 [*DOI :* 10.1007/978-3-030-21485-2_19], https://hal.inria.fr/hal-02420485

[15] I. FOUAD, N. BIELOVA, A. LEGOUT, N. SARAFIJANOVIC-DJUKIC. *Missed by Filter Lists: Detecting Unknown Third-Party Trackers with Invisible Pixels*, in "PETS 2020 - 20th Privacy Enhancing Technologies Symposium", Montréal, Canada, PETs (Privacy Enhancing Technologies Symposium), July 2020, https://hal.inria.fr/hal-01943496

[16] B. PETIT, M. SERRANO. *Composing and Performing Interactive Music using the HipHop.js language*, in "NIME 2019 - New Interfaces for Musical Expression", Porto Allegre, Brazil, June 2019, https://hal.inria.fr/hal-02410197

[17] M. SERRANO, M. FEELEY. *Property caches revisited*, in "CC 2019 - 28th International Conference on Compiler Construction", Washington, United States, ACM Press, February 2019, pp. 99-110 [*DOI :* 10.1145/3302516.3307344], https://hal.inria.fr/hal-02418678

[18] D. F. SOMÉ. *EmPoWeb: Empowering Web Applications with Browser Extensions*, in "SP 2019 - 40th IEEE Symposium on Security and Privacy", San Francisco, United States, May 2019, https://hal.archives-ouvertes.fr/hal-02433525

Scientific Books (or Scientific Book chapters)

[19] I. CASTELLANI, M. R. MOUSAVI. *Special Issue on Trends in Concurrency Theory (selected invited contributions from the workshops TRENDS 2015 and 2016)*, Elsevier, October 2019, vol. 107, pp. 175-176 [*DOI :* 10.1016/J.JLAMP.2019.07.001], https://hal.inria.fr/hal-02422352

Research Reports

[20] M. CANO, I. CASTELLANI, C. DI GIUSTO, J. A. PÉREZ. *Multiparty Reactive Sessions*, Inria, April 2019, n^{o} 9270, 65 p. , https://hal.archives-ouvertes.fr/hal-02106742

[21] I. CASTELLANI, M. DEZANI-CIANCAGLINI, P. GIANNINI. *Event structure semantics for multiparty sessions*, INDES, March 2019, n^{o} RR-9266, https://hal.inria.fr/hal-02081943

References in notes

[22] A. CORMACK. *Is the Subject Access Right Now Too Great a Threat to Privacy?*, in "European Data Protection Law Review", 2016, vol. 2, n^{o} 1, pp. 15-27