Verified cryptographic security for Web applications

Mots clés :
● Directeur de thèse : Michel BERA
● Co-encadrant(s) :
● Unité de recherche : Laboratoire inconnu!
● Ecole doctorale : École Doctorale Informatique, Télécommunications, Électronique de Paris
● Domaine scientifique principal: Divers

Résumé du projet de recherche (Langue 1)

Web applications such as online banking, cloud-based storage and electronic voting must handle and manipulate sensitive user data distributed and replicated across various clients and servers, leading to common concerns about user privacy and data confidentiality. To protect against malicious websites and compromised machines, these applications rely on isolation mechanisms provided by web browsers, on authentication and authorization protocols, on application-level cryptographic constructions, and on transport-layer cryptographic protocols. However, combining these varied mechanisms to achieve high-level security goals is error-prone, and has led to many attacks even on sophisticated and well-studied applications. The goal of the proposed dissertation is to design and implement a programming and verification framework that enables developers to write web applications with provable privacy for user data. We will investigate and formalize various notions of expected privacy guarantees and evaluate whether existing web application designs and enforcement mechanisms are capable of meeting these goals. We will propose formal models for new HTML5 security mechanisms. We will design and implement new cryptographic mechanisms that can provide strong security even when some clients and servers are compromised. We will develop program analyses that can verify that a web application satisfies a formal privacy goal. In combination, we seek to develop security libraries and web applications that are the first to offer provable privacy against a strong attacker model that combines web attackers, network attackers, and compromised hosts. Our focus is on web applications that are written in JavaScript and executed within HTML5 web browsers on the client and within frameworks like Node on the server. Our work will build on previous security type systems for JavaScript [Bhargavan et al 2013a, Swamy et al 2014], and on the mechanised formal semantics of JavaScript [Maffeis et al..]. For transport-level security, we will rely on and extend a previously verified implementation of the TLS protocol [Bhargavan et al 2013a, 2008]. Our applications and case studies will be drawn from existing and new cryptographic web applications such as Cryptocat, Persona, LastPass, and Helios.