A SAT-Based Planning Approach for Finding Logical Attacks on Cryptographic Protocols

Noureddine Aribi, LITIO Lab., Université Oran1, Oran, Algeria
Yahia Lebbah, LITIO Lab., Université Oran1, Oran, Algeria

ABSTRACT

Cryptographic protocols form the backbone of digital society. They are concurrent multiparty communication protocols that use cryptography to achieve security goals such as confidentiality, authenticity, integrity, etc., in the presence of adversaries. Unfortunately, protocol verification still represents a critical task and a major cost to engineer attack-free security protocols. Model checking and SAT-based techniques proved quite effective in this context. This article proposes an efficient automatic model checking approach that exemplies a security property violation. In this approach, a protocol verification is abstracted as a compact planning problem, which is efficiently solved by a state-of-the-art SAT solver. The experiments performed on some real-world cryptographic protocols succeeded in detecting new logical attacks, violating some security properties. Those attacks encompass both “type flaw” and “replay” attacks, which are difficult to tackle with the existing planning-based approaches.

KEYWORDS

Cryptography, Formal Verification, Logical Attacks, Model Checking, PDDL, Planning, SAT Solvers, Security Protocols

INTRODUCTION

Security protocols are short concurrent communication programs designed to secure exchanges over a network, using encryption primitives. They are intended to achieve various goals depending on the target application, e.g. confidentiality, authenticity, and integrity. Their verification has shown its interest for detecting and preventing different types of attacks, e.g., eavesdropping, replay attack, password guessing attack, man-in-the-middle attack and quantum computer attack (Paul, Kumar, & Suman, 2019). Cryptographic protocols are ubiquitous and widely used in applications that require secure electronic transactions, such as banking, e-voting, e-commerce, e-mail exchange, and so on. They are also increasingly designed and used to secure new emerging technologies like the Internet of Things (IoT) (Kaur, Verma, Jain, & Kesswani, 2019), RFID tags, mobile phones, wireless sensors

DOI: 10.4018/IJISP.2020100101

Copyright © 2020, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
and Wi-Fi networks (Rath & Pattanayak, 2019). Cryptography appears here as an essential tool to ensure security properties, in particular by encryption algorithms and hash functions. Although security protocols are typically small, analysis by hand is an extremely challenging, error prone and time-consuming problem, since a protocol should work even when many runs are interleaved and in the presence of a malicious adversary. Failures in such protocols may lead to financial loss, violations of personal privacy, and real threats to the economy (Ben, 2011).

Recent model checking and automated verification approaches have made it possible to find and build attacks over protocols, which had apparently been undiscovered for many years. A famous example is the Needham-Schroeder protocol (Needham & Schroeder, 1978) on which Lowe (1995) has discovered a flaw 17 years after its publication. Hence, verification approaches are proposed to help protocol designers in writing, adjusting and strengthening the security protocol specification (Cramer, Damgard, & Nielsen, 2015; Schoenmakers, 2019). This article advances verification techniques for cryptographic protocols using formal methods with an emphasis on expressiveness, compactness and automation.

Let us briefly review some existing approaches used to analyze security protocols. Methods based on induction and theorem proving have been developed in the literature (Cortier, Kremer, & Warinschi, 2011; Paulson, 1997). Those methods ensure correctness proofs and they can handle protocols by considering several unbounded dimensions (unbounded number of parallel and sequential runs, unbounded number of principals, nonces, and encryption keys). However, unlike model checking, they are difficult to implement and not fully automated (require user interaction). Moreover, when the security verifier fails, it is difficult to build counter-examples violating some security properties. Other methods such as first-order resolution address the verification problem by attempting to prove correctness on unbounded protocols by means of the Horn formal logic language, which includes an inference procedure as well as suitable proof strategies. Those methods are automatic and successfully implemented using Prolog programs (Blanchet, 2001). However, the termination of their analysis is not guaranteed in general. In addition, protocol abstraction may lead to false positive results (false attacks).

Alternatively, model-checking techniques offer a cheaper alternative to formal correctness proofs (Basin, Cremers, & Meadows, 2018), by focusing on counter-examples extraction that reveal a security failure. Thus, instead of proving a protocol attack-free, a model checker automatically explores the state space of a security model in an attempt to prove a potential security vulnerability (a.k.a. a logical attack). Despite the search space explosion of the model-checking approach, it remains more convenient than methods based on pure mathematical proofs, especially once combined with recent Prepositional Satisfiability (SAT) solving techniques. In this way, many model-checking tools have been developed to achieve finite-state analysis such as FDR (Donovan, Norris, & Lowe, 1999), Mur $\varphi$ (Mitchell, Mitchell, & Stern, 1997), Brutus (Clarke, Jha, & Marrero, 2000) and SATMC (Armando, Carbone, & Compagna, 2014). To the best of our knowledge, Compagna (2005) was the first pioneer in exploiting a planning approach to analyze cryptographic protocols. Its approach operates in three main steps summarized below:

1. Using the AVISPA tool to translate a protocol insecurity problem (PIP) into a state transition system written in an intermediate format (IF);
2. Converting the PIP into a planning problem specified in SATE language; and finally;
3. Generating a CNF (Conjunctive Normal Form) formula to be solved with state-of-the-art SAT solvers.

Instead, this paper proposes a two steps model checking approach that improves the verification approach just described with the following main contributions:
The Compliance of IT Control and Governance: A Case of Macao Gaming Industry
[www.igi-global.com/article/the-compliance-of-it-control-and-governance/155103?camid=4v1a](www.igi-global.com/article/the-compliance-of-it-control-and-governance/155103?camid=4v1a)

The Impact of Privacy Risk Harm (RH) and Risk Likelihood (RL) on IT Acceptance: An Examination of a Student Information System
[www.igi-global.com/chapter/impact-privacy-risk-harm-risk/30107?camid=4v1a](www.igi-global.com/chapter/impact-privacy-risk-harm-risk/30107?camid=4v1a)