AnBx - Security Protocols Design and Verification*

Michele Bugliesi and Paolo Modesti

Università Ca' Foscari Venezia
Dipartimento di Informatica
{bugliesi,modesti}@dsi.unive.it

Abstract. Designing distributed protocols is challenging, as it requires actions at very different levels: from the choice of network-level mechanisms to protect the exchange of sensitive data, to the definition of structured interaction patterns to convey application-specific guarantees. Current security infrastructures provide very limited support for the specification of such guarantees. As a consequence, the high-level security properties of a protocol typically must often be hard-coded explicitly, in terms of low-level cryptographic notions and devices which clutter the design and undermine its scalability and robustness.

To counter these problems, we propose an extended *Alice & Bob* notation for protocol narrations (AnBx) to be employed for a purely declarative modelling of distributed protocols. These abstractions provide a compact specification of the high-level security guarantees they convey, and help shield the design from the details of the underlying cryptographic infrastructure. We discuss an implementation of the abstractions based on a translation from the AnBx notation to the AnB language supported by the OFMC [1,2] verification tool. We show the practical effectiveness of our approach by revisiting the iKP e-payment protocols, and showing that the security goals achieved by our declarative specification outperform those offered by the original protocols.

1 Introduction

On-line transactions represent an important share of the overall world trade and security constitutes a major concern in these kind of applications, as agreeing, on the terms of a transaction in a distributed and open environment like the internet, requires protection against threats from intruders and/or from the potential misbehavior of other participants. Establishing the desired safeguards is challenging as it involves actions at different levels: from the choice of core, network-level mechanisms to protect the exchange of sensitive data, to the definition of structured, application-specific measures to enforce the high-level behavioral invariants of the participants. Current security infrastructures offer effective abstractions only for the core mechanisms, based on tools such as TLS/SSL [3]

* Work partially supported by MIUR Projects SOFT "*Security Oriented Formal Techniques*" and IPODS "*Interacting Processes in Open-ended Distributed Systems*".

A. Armando and G. Lowe (Eds.): ARSPA-WITS 2010, LNCS 6186, pp. 164–184, 2010.

to provide tunneling support for communication. On the other hand, little to no support is provided for the specification of more structured interaction patterns, so that high-level security invariants must typically be expressed, and hard-coded explicitly, in terms of low-level cryptographic notions such as salting, nonces, keyed-hashing, encryptions, signature schemes, and compositions thereof. As a result, the application code and data structures get intertwined with low-level code that not only gets in the way of a clear understanding of the applications' business logic, but also undermines its scalability and robustness.

To counter these problems, various papers in the recent literature (see, e.g., [4,5,6]) have advocated a programming discipline based on (*i*) high-level security abstractions and mechanisms for composing them to support structured interaction patterns [7,8], and (*ii*) automatic techniques to build defensive implementations on top of well-established cryptographic infrastructures and tools.

Following this line of research, in the present paper we isolate a core set of channel and data abstractions to be employed for a purely declarative modelling of distributed protocols. Our abstractions are part of `AnBx`, a dialect of the well-known *Alice & Bob* (AnB) notation for protocol narrations, which supports various mechanisms for securing remote communications based on abstract security *modes*, without any reference to explicit cryptography. The `AnBx` abstractions are readily translated into corresponding public-key cryptographic protocols described by standard AnB narrations. This provides an abstract, yet effective implementation of the `AnBx` specification language, to be employed as the basis for the development of fully-fledged implementation.

Main contributions and results. We developed a compiler for the automatic translation from `AnBx` to the AnB notation used in the symbolic model-checker OFMC (Open-source Fixed-point Model Checker [1,2]), and verified the soundness of our implementation with OFMC itself. The translation allows the verification of `AnBx` protocols with any OFMC-interoperable verification tool [9]. To experiment and validate the practical effectiveness of the `AnBx` approach to protocol design, we revisited the *iKP* e-payment protocol family (Internet Keyed Payment Protocol [10,11]) as a case study, and contrasted the security goals achieved by our version with those offered by the original protocol.

Interestingly, our `AnBx` versions of the iKP protocols outperform the original protocols (for all i's), i.e. they satisfy stronger security goals and properties. This is largely a consequence of the declarative nature of the specification style supported by `AnBx`: being defined as channel-level abstractions, the `AnBx` primitives convey protection on *all* message components, not just on some components as in the original iKP specification, yielding stronger encapsulation mechanisms, and consequently, stronger and more scalable security guarantees. As a byproduct of our comparative analysis, we also found a (to the best of our knowledge) new flaw in the original specification of {2,3}KP, and proposed an amended version that rectifies the problem.

Plan of the paper. In Section 2 we introduce the `AnBx` specification language together with our high-level security abstractions; in Section 3 we outline a