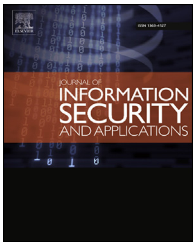
Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/jisa



Security protocol specification and verification with *AnBx*

Michele Bugliesi [a], *Stefano Calzavara* [a], *Sebastian Mödersheim* [b], *Paolo Modesti* [c,*,1]

[a] *Dipartimento di Scienze Ambientali Informatica e Statistica, Università Ca' Foscari Venezia, Venezia, Italy*
[b] *DTU Compute, Danmarks Tekniske Universitet, Kgs. Lyngby, Denmark*
[c] *School of Computing Science, Newcastle University, Newcastle upon Tyne, UK*

ARTICLE INFO

Article history:
Available online 21 June 2016

Keywords:
Protocol specification
Protocol verification
Model-checking
e-payment

ABSTRACT

Designing distributed protocols is complex and requires actions at very different levels: from the design of an interaction flow supporting the desired application-specific guarantees to the selection of the most appropriate network-level protection mechanisms. To tame this complexity, we propose *AnBx*, a formal protocol specification language based on the popular *Alice & Bob* notation. *AnBx* offers channels as the main abstraction for communication, providing different authenticity and/or confidentiality guarantees for message transmission. *AnBx* extends existing proposals in the literature with a novel notion of *forwarding* channels, enforcing specific security guarantees from the message originator to the final recipient along a number of intermediate forwarding agents. We give a formal semantics of *AnBx* in terms of a state transition system expressed in the AVISPA Intermediate Format. We devise an ideal channel model and a possible cryptographic implementation, and we show that, under mild restrictions, the two representations coincide, thus making *AnBx* amenable to automated verification with different tools. We demonstrate the benefits of the declarative specification style distinctive of *AnBx* by revisiting the design of two existing e-payment protocols: *iKP* and *SET*.

1. Introduction

The *Alice & Bob* notation, also known as *protocol narrations*, is a popular device which has been widely adopted in the literature as the basis of several security protocol specification frameworks (Almousa et al., 2015; Chevalier and Rusinowitch, 2010; Denker et al., 2000; Jacquemard et al., 2000; Lowe, 1998; Mödersheim, 2009). In such frameworks, the semantics of the specification languages is defined by a translation into lower level formats, amenable to model-checking and automated verification. Besides making verification possible, the translation semantics provides for a clean separation between the abstract specification of the protocol structure and the details of its implementation, which may be generated directly from the specification (Almousa et al., 2015; Carlsen, 1994; Jakobsson et al., 1996; Millen and Muller, 2001; Modesti, 2014, 2016; Quaresma and Probst, 2010). This separation has a beneficial impact on both the specification and the implementation: on the one hand, it helps focusing on application-level properties,

[*] *Corresponding author.* Department of Computing, Engineering and Technology, University of Sunderland, St. Peters Way, Sunderland SR60DD, UK. Tel.: +44-191-515-2382; fax: +44-191-515-2781.
E-mail address: paolo.modesti@sunderland.ac.uk (P. Modesti).
[1] Present address: Department of Computing, Engineering and Technology, University of Sunderland, Sunderland, UK.

http://dx.doi.org/10.1016/j.jisa.2016.05.004
2214-2126/© 2016 Elsevier Ltd. All rights reserved.

staying away from unnecessary low-level details; on the other hand, it contributes to strengthening the implementation and to ensure the protocol end-to-end security, by delegating to the compiler the selection of the most adequate core implementation components.

Channel abstractions make a further step in the same direction: they help in designing distributed applications irrespective of the cryptographic mechanisms needed to protect communication, by interpreting channels as a secure communication medium with built-in protection against certain attacks (e.g., on confidentiality).

How these properties are actually ensured represents a different design aspect, which might not be a concern of the application designer at all, and may be left to the compiler.

1.1. Related work

Several papers in the literature have taken this approach, and developed it along different directions. First, there are papers that propose the definition and implementation of different channel abstractions, based on cryptographic realizations and interaction patterns. Abadi et al. (2000) propose a process calculus with native constructs for authentication and discuss a possible cryptographic implementation. Adao and Fournet (2006) design a variant of the pi-calculus with secure communication and describe its computationally sound compilation into a concrete implementation. Other authors explore the idea of compiling secure protocols for distributed sessions from convenient ML abstractions based on session types, a powerful formalism used to structure interaction and reason over communicating processes and their behaviour (Bhargavan et al., 2009; Corin et al., 2007).

Another line of research, instead, is more focused on reasoning about channels and their ideal behaviour in an abstract way. Dilloway and Lowe (2007) present a hierarchy of secure channels and discuss their relative strengths. Bugliesi and Focardi (2008) devise secure channel abstractions in a process algebraic setting and reason about the relative power of a low-level adversary. Armando et al. (2007) model different channel types using set-rewriting and linear temporal logic. Kamil and Lowe (2009, 2011) adapt the Strand Spaces model to deal with secure channels, providing different security guarantees.

Mödersheim and Viganò (2009) consider both an abstract characterization and a concrete realization of channels, showing that both characterizations coincide; the paper defines also the notion of channels as goals and proves a related compositionality result. The same authors also formalize some easy-to-check static conditions that support a large class of channels and applications and that are sufficient for vertical security protocol composition (Mödersheim and Viganò, 2014). These works also demonstrated that Alice and Bob notation is ideal for the combination with the channel notation, and channel types were integrated both in the languages AnB (Mödersheim, 2009) and SPS (Almousa et al., 2015). In these papers, the focus is on giving a very general and concise semantics to Alice and Bob notation, namely defining with a few mathematically simple principles the semantics in presence of an arbitrary algebraic theory. With respect to this semantics, Almousa et al. (2015) prove the correctness of a translator to formal models and implementations. Our paper is based on this semantic machinery for the cryptographic handling of messages and defines a rich set of channels on top of this basis.

We should mention two more related works on channels. Gibson-Robinson (2013) employs the notion of channel (and their properties) for the analysis of multi-layer security protocols. Finally, Sprenger and Basin (2010, 2012) consider a refinement approach where cryptographic protocols are synthesised from high-level security goals; one of the steps of the refinement process builds on the usage of channel abstractions.

1.2. Contributions

In the present paper we develop channels one step further, generalizing them to capture the notion of *forwarding channel*, a critical abstraction for designing and reasoning about complex protocols involving three or more communicating parties. A typical scenario for such protocols is represented by e-commerce transactions, in which a customer requires a merchant to certify that her payment has been cleared out, and the merchant provides that evidence by forwarding to the customer the notification she received from the credit card issuer. Similarly, single sign-on protocols usually involve an authenticity-preserving forwarding of access tokens from a trusted third-party to different clients. This kind of interactions may be modelled by session types, since they are typically developed on top of very expressive calculi and languages, but it is not accounted for in existing protocol narration frameworks with channel abstractions. Including forwarding in these frameworks is important, given their wide popularity and ease of use.

We develop the novel concept of forwarding channel as part of *AnBx*, a formal specification language that we introduce by conservatively extending the semantics of the *AnB* language (Mödersheim, 2009). *AnBx* includes modes for all kinds of message forwarding, where all or some of the properties of the original transmission are preserved upon relaying. In our characterization, we provide both an abstract interpretation of channels that captures their ideal behaviour and a cryptographic implementation, and we prove a formal equivalence between the two characterizations. Both interpretations are based on a translation to the AVISPA Intermediate Format, hence *AnBx* is directly available for automated verification with the different tools that use this format, such as OFMC (Basin et al., 2005).

We demonstrate the practical effectiveness of our approach by an analysis and re-engineering of two real-life e-payment protocols: *iKP* [Internet Keyed Payment (Bellare et al., 1995, 2000)] and *SET* [Secure Electronic Transaction (Bella et al., 2003, 2005, 2006)]. Although both protocols could be expressed in their full complexity in *AnBx*, we rely on the abstract channels available in the language to factor out the cryptographic aspects almost entirely. The resulting protocols are more concise, easier to understand and, interestingly, more efficient to verify than the original versions.

In addition, the *AnBx* formulations strengthen the original specifications in that they enjoy stronger security goals and properties. As a by-product of our comparative analysis, we also find a (to the best of our knowledge) new flaw in the original specification of *iKP* and propose an amended version that rectifies the problem.