Extending Session Types to Model Security Properties

Julie Tollund

4th December 2018

Contents

1 Introduction 1

1.1 Motivation 1

1.2 Current research 1

1.3 Intended outcome 1

2 Work Done 3

2.1 Applied π-Calculus 3

2.1.1 Terms 3

2.1.2 Examples 4

2.2 Security Protocols 4

2.2.1 Examples 5

2.3 Session Types (and choreography programming) 5

2.4 TPM 5

2.5 Evaluation 6

3 Future Plan 7

1 | Introduction

The purpose of this report, is to establish a foundation for a future thesis by the author, in regards of extending session types to model security properties with the introduction of adversaries.

In the report I will first introduce the motivation for exploring this field, as well as the current research done in the area. Secondly, I will introduce different research areas, all related to the field, and how these work together as background knowledge for the further thesis, by which I will explain further about in the final section of this report.

1.1 Motivation

With IT becoming an ever bigger part of our lives, the need for stronger and better security measurements, has grown with it.
Voting machines
Confidentiality
Ever expanding field
Tighter restrictions (session types - adversaries)
TPM (new security measurement)

1.2 Current research

The project relies heavily on the research done within the field of Security protocols and Session types.
Mark Ryan's research on the TPM (citation).
ProVerif?

1.3 Intended outcome

"The intended outcome of the thesis, is to take the idea of session type and choreography programming, and consider them in an adversarial environment. This will be done by extending session types to model properties, and from this produce protocols that are secure by construction." (Taken from Thesis prep agreement). Compiler.

2 | Work Done

This section describes the work carried out so far (rephrase). Most of it will be highlighting research done within the different fields, and showcasing examples of its use. The section ends in a summery of how the different fields come together, and how they can be used further in the coming thesis.

2.1 Applied π-Calculus

The applied pi-calculus (ref. Abadi and Fournet, 2001) is based upon the language pi-calculus, but offers a more convenient use for modelling security protocols to be specified, by allowing for a more wide variety of complex primitives. It is used for describing and analysing security protocols, as it provides a more intuitive process syntax for detailing the actions of the participants in a protocol [1]. This is done by introducing a rich term algebra, for modelling the cryptographic operations used in security protocols, where function symbols represent cryptographic protocols.

Tools such as ProVerif [9], uses a syntax closely related to the applied pi-calculus, and offers a way of automated reasoning about the security properties found in cryptographic protocols.

2.1.1 Terms

As mentioned, the applied pi-calculus uses Terms (in contrast to just names in the pi-calculus) to model messages that are exchanged during a protocol. Terms are built over a signature, names and variables[1]:

L, M, N, T, U, V ::=	terms
a, b, c,...,k,...,m, n,..,s	names
x, y, z	variables
g(M$_1$,..,M$_l$)	function application

2.1.2 Examples

Having established an understanding of how the the applied pi-calculus differentiate from the pi-calculus, we can now look at how some of the known protocols can be described.
First of, we look at the simple Handshake protocol used for setting the parameters for communication between two devices, such as the old dial-up modem or when connecting to a USB.
Handshake protocol with applied pi-calculus as illustred by Ryan and Smyth:

$$\begin{aligned} P \quad &\triangleq \quad \nu\, sk_{\mathcal{S}}.\nu\, sk_{\mathcal{C}}.\nu\, s. \\ &\quad\ \ \text{let } pk_S = \mathsf{pk}(sk_{\mathcal{S}}) \text{ in let } pk_C = \mathsf{pk}(sk_{\mathcal{C}}) \text{ in} \\ &\quad\ \ (\overline{c}\langle pk_S\rangle \mid \overline{c}\langle pk_C\rangle \mid !P_{\mathcal{S}} \mid !P_{\mathcal{C}}) \end{aligned}$$

$$\begin{aligned} P_{\mathcal{S}} \quad &\triangleq \quad c(x\_pk).\nu\, k.\overline{c}\langle \mathsf{aenc}(x\_pk, \mathsf{sign}(sk_{\mathcal{S}}, k))\rangle. \\ &\quad\ \ c(z).\text{if } \mathsf{fst}(\mathsf{sdec}(k, z)) = tag \text{ then } Q \end{aligned}$$

$$\begin{aligned} P_{\mathcal{C}} \quad &\triangleq \quad c(y).\text{let } y' = \mathsf{adec}(sk_{\mathcal{C}}, y) \text{ in let } y\_k = \mathsf{getmsg}(y') \text{ in} \\ &\quad\ \ \text{if } \mathsf{checksign}(pk_S, y') = \mathsf{true} \text{ then} \\ &\quad\ \ \overline{c}\langle \mathsf{senc}(y\_k, \mathsf{pair}(tag, s))\rangle \end{aligned}$$

Needham-Schroeder Public key protocol with applied pi-calculus by Cortier and Kremer (Describe P$_A$ and P$_B$ first):

$$\begin{aligned} P^1_{\mathsf{nspk}} \hat{=} \nu sk_a, sk_b.(P_A(sk_a, \mathsf{pk}(sk_b)) \parallel P_B(sk_b) \parallel \\ \mathsf{out}(c, \mathsf{pk}(sk_a)) \parallel \mathsf{out}(c, \mathsf{pk}(sk_b))) \end{aligned}$$

TODO: Do description and explanation of both

2.2 Security Protocols

Security protocols is an abstract or concrete protocol, that characterise the security related functions and applies cryptographic methods. It describes how the algorithm should be used to ensure the security and integrity of data transmitted. The security protocol is a protocol that runs in an untrusted environment, where it usually assumes channels are untrusted and participants are dishonest. In academic examples, they are often described with the Alice and Bob notation. (The Dolev-Yao model)

A way of reason about wether a message is deductible by an adversary is through inference rules. Inference rules offers a formal analysis for proving security properties of protocols.

TODO: discuss how Inference rules and derivation sequences can be used

2.2.1 Examples

TODO: show examples of a man in the middle attack on the NS and DH protocols:

- Needham-Schroeder Public key protocol (now modified according to Lowe's man in the middle attack):

$$\begin{aligned} P^5_{\mathsf{nspk}} \mathrel{\hat{=}} !\nu sk_a, sk_b.(&!\mathsf{in}(c, x_{pk}).P_A(sk_a, x_{pk}) \parallel !P_B(sk_a) \parallel \\ &!\mathsf{in}(c, x_{pk}).P_A(sk_b, x_{pk}) \parallel !P_B(sk_b) \parallel \\ &\mathsf{out}(c, \mathsf{pk}(sk_a)) \parallel \mathsf{out}(c, \mathsf{pk}(sk_b))) \end{aligned}$$

- Diffie-Hellman protocol (man in the middle attack)

2.3 Session Types (and choreography programming)

- Description of session types and what they are used for
- Research done within the field
- Global session types
- Examples?

2.4 TPM

- Description of the TPM - what it is, and what it's used for
- Examples of TPM commands with applied pi-calculus

The Trusted Platform Module (TPM) is a specialised chip that stores RSA encryption keys specific for the host system for hardware authentication. It is used as a component on an endpoint device and is used for the Windows BitLocker. The TPM contains an RSA key pair called the Endorsement Key (EK), together with an owner-specified password. The Storage Root Key (SRK) is created when a user or administrator takes ownership of the system (rephrase), and works in a tree like structure to store TPM generated RSA keys used.

The TPM offers two authorisation sessions:

- Object Independent Authorisation Protocol (OIAP)
- Object Specific Authorisation Protocol (OSAP)

The OIAP creates a session that can manipulate any object, but will only work with certain commands. The OSAP creates a session that can only manipulate a specific object, specified at the session start.

2.5 Evaluation

- How they all come together
- Examples of TPM commands with session types

3 | Future Plan

Farewell

References

[1] M. D. Ryan and B. Smyth, “Applied pi calculus”, 2010.

[2] A. Mukhamedov, A. D. Gordon and M. Ryan, “Towards a verified reference implementation of a trusted platform module”, in *Security Protocols XVII, 17th International Workshop, Cambridge, UK, April 1-3, 2009. Revised Selected Papers*, 2009.

[3] S. Gürgens, C. Rudolph, D. Scheuermann, M. Atts and R. Plaga, “Security evaluation of scenarios based on the tcg’s TPM specification”, in *Computer Security - ESORICS 2007, 12th European Symposium On Research In Computer Security, Dresden, Germany, September 24-26, 2007, Proceedings*, 2007.

[4] S. Delaune, S. Kremer, M. D. Ryan and G. Steel, “A formal analysis of authentication in the TPM”, in *Formal Aspects of Security and Trust - 7th International Workshop, FAST 2010, Pisa, Italy, September 16-17, 2010. Revised Selected Papers*, 2010.

[5] H. Hüttel, I. Lanese, V. T. Vasconcelos, L. Caires, M. Carbone, P. Deniélou, D. Mostrous, L. Padovani, A. Ravara, E. Tuosto, H. T. Vieira and G. Zavattaro, “Foundations of session types and behavioural contracts”, *ACM Comput. Surv.*, vol. 49, no. 1, 3:1–3:36, 2016.

[6] V. T. Vasconcelos, “Fundamentals of session types”, *Inf. Comput.*, vol. 217, pp. 52–70, 2012.

[7] V. Cortier and S. Kremer, “Formal models and techniques for analyzing security protocols: A tutorial”, *Foundations and Trends in Programming Languages*, vol. 1, no. 3, pp. 151–267, 2014.

[8] S. Delaune, S. Kremer, M. D. Ryan and G. Steel, “Formal analysis of protocols based on TPM state registers”, in *Proceedings of the 24th IEEE Computer Security Foundations Symposium, CSF 2011, Cernay-la-Ville, France, 27-29 June, 2011*, 2011, pp. 66–80.

Online references

[9] B. Blanchet, *Proverif*. [Online]. Available: `http://prosecco.gforge.inria.fr/personal/bblanche/proverif/` (visited on 03/12/2018).