# Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif

Full text available at: http://dx.doi.org/10.1561/3300000004

# Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif

**Bruno Blanchet**
INRIA Paris, France
Bruno.Blanchet@inria.fr

the essence of knowledge
Boston — Delft

# Foundations and Trends® in Privacy and Secruity

*Published, sold and distributed by:*
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

*Outside North America:*
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

B. Blanchet. *Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif*. Foundations and Trends® in Privacy and Secruity, vol. 1, no. 1-2, pp. 1–135, 2016.

ISBN: 978-1-68083-207-5

# Foundations and Trends® in Privacy and Secruity

Volume 1, Issue 1-2, 2016

## Editorial Board

# Editorial Scope

## Topics

Foundations and Trends® in Privacy and Security publishes survey and tutorial articles in the following topics:

- Access control
- Accountability
- Anonymity
- Application security
- Artifical intelligence methods in security and privacy
- Authentication
- Big data analytics and privacy
- Cloud security
- Cyber-physical systems security and privacy
- Distributed systems security and privacy
- Embedded systems security and privacy
- Forensics
- Hardware security
- Human factors in security and privacy
- Information flow
- Intrusion detection
- Malware
- Metrics
- Mobile security and privacy
- Language-based security and privacy
- Network security
- Privacy-preserving systems
- Protocol security
- Security and privacy policies
- Security architectures
- System security
- Web security and privacy

## Information for Librarians

Foundations and Trends® in Privacy and Security, 2016, Volume 1, 4 issues. ISSN paper version 2474-1558. ISSN online version 2474-1566. Also available as a combined paper and online subscription.

# Contents

# Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif

Bruno Blanchet[1]

[1] *INRIA Paris, France; Bruno.Blanchet@inria.fr*

ABSTRACT

ProVerif is an automatic symbolic protocol verifier. It supports a wide range of cryptographic primitives, defined by rewrite rules or by equations. It can prove various security properties: secrecy, authentication, and process equivalences, for an unbounded message space and an unbounded number of sessions. It takes as input a description of the protocol to verify in a dialect of the applied pi calculus, an extension of the pi calculus with cryptography. It automatically translates this protocol description into Horn clauses and determines whether the desired security properties hold by resolution on these clauses. This survey presents an overview of the research on ProVerif.

Bruno Blanchet (2016), "Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif", Foundations and Trends® in Privacy and Secruity: Vol. 1, No. 1-2, pp 1–135. DOI: 10.1561/3300000004.

# 1

# Introduction

## Verifying security protocols

The verification of security protocols has been an active research area since the 1990s. This topic is interesting for several reasons. Security protocols are ubiquitous: they are used for e-commerce, wireless networks, credit cards, e-voting, among others. The design of security protocols is notoriously error-prone. This point can be illustrated by attacks found against many published protocols. For instance, a famous attack was discovered by Lowe (1996) against the Needham-Schroeder public-key protocol (Needham and Schroeder, 1978) 17 years after its publication. Attacks are also found against many protocols used in practice. Important examples are SSL (Secure Sockets Layer) and its successor TLS (*Transport Layer Security*), which are used for https:// connexions. The first version dates back to 1994, and since then many attacks were discovered, fixed versions were developed, and new attacks are still regularly discovered (Beurdouche *et al.*, 2015; Adrian *et al.*, 2015). Moreover, security errors cannot be detected by functional testing, since they appear only in the presence of a malicious adversary. These errors can also have serious consequences. Hence, the formal verification or proof of protocols is particularly desirable.

### Modeling security protocols

In order to verify protocols, two main models have been considered:

- In the *symbolic model*, often called Dolev-Yao model and due to Needham and Schroeder (1978) and Dolev and Yao (1983), cryptographic primitives are considered as perfect blackboxes, modeled by function symbols in an algebra of terms, possibly with equations. Messages are terms on these primitives and the adversary can compute only using these primitives. This is the model usually considered by formal method practitioners.
- In contrast, in the *computational model*, messages are bitstrings, cryptographic primitives are functions from bitstrings to bitstrings, and the adversary is any probabilistic Turing machine. This is the model usually considered by cryptographers.

The symbolic model is an abstract model that makes it easier to build automatic verification tools, and many such tools exist: AVISPA (Armando *et al.*, 2005), FDR (Lowe, 1996), Scyther (Cremers, 2008), Tamarin (Schmidt *et al.*, 2012), for instance. The computational model is closer to the real execution of protocols, but the proofs are more difficult to automate; we refer the reader to (Blanchet, 2012a) and to Chapter 6 for some information on the mechanization of proofs in the computational model.

Most often, the relations between cryptographic primitives given in the symbolic model also hold in the computational model.[1] In this case, an attack in the symbolic model directly leads to an attack in the computational model, and a practical attack. However, the converse is not true in general: a protocol may be proved secure in the symbolic model, and still be subject to attacks in the computational model. For this reason, the *computational soundness* approach was introduced: it proves general theorems showing that security in the symbolic model implies security in the computational model, modulo additional assumptions. However, since the two models do not coincide, this approach

[1] Sometimes, one may also overapproximate the capabilities of the adversary in the symbolic model.

typically requires strong assumptions on the cryptographic primitives (for instance, encryption has to hide the length of the messages) and on the protocol (for instance, absence of key cycles, in which a key is encrypted under itself; correctly generated keys, even for the adversary). This approach was pioneered by Abadi and Rogaway (2002). This work triggered much research in this direction; we refer to (Cortier *et al.*, 2011) for a survey.

Even though the computational model is closer to reality than the symbolic model, we stress that it is still a model. In particular, it does not take into account side channels, such as timing and power consumption, which may give additional information to an adversary and enable new attacks. Moreover, one often studies specifications of protocols. New attacks may appear when the protocol is implemented, either because the specification has not been faithfully implemented, or because the attacks rely on implementation details that do not appear at the specification level.

In this survey, we focus on the verification of specifications of protocols in the symbolic model. Even though it is fairly abstract, this level of verification is relevant in practice as it enables the discovery of many attacks.

### Target security properties

Security protocols can aim at a wide variety of security goals. The main security properties can be classified into two categories, *trace properties* and *equivalence properties*. We define these categories and mention two particularly important examples: secrecy and authentication. These are two basic properties required by most security protocols. Some protocols, such as e-voting protocols (Delaune *et al.*, 2009), require more complex and specific security properties, which we will not discuss.

### Trace and equivalence properties

Trace properties are properties that can be defined on each execution trace (each run) of the protocol. The protocol satisfies such a property when it holds for all traces. For example, the fact that some states are unreachable is a trace property.

Equivalence properties mean that the adversary cannot distinguish two processes (that is, protocols). For instance, one of these processes can be the protocol under study, and the other one can be its specification. Then, the equivalence means that the protocol satisfies its specification. Therefore, equivalences can be used to model many subtle security properties. Several variants exist (observational equivalence, testing equivalence, trace equivalence) (Abadi and Gordon, 1999; Abadi and Gordon, 1998; Abadi and Fournet, 2001). Observational equivalence provides compositional proofs: if a protocol $P$ is equivalent to $P'$, $P$ can be replaced with $P'$ in a more complex protocol. However, the proof of equivalences is more difficult to automate than the proof of trace properties: equivalences cannot be expressed on a single trace, they require relations between traces (or processes).

### Secrecy

Secrecy, or confidentiality, means that the adversary cannot obtain some information on data manipulated by the protocol. Secrecy can be formalized in two ways:

- Most often, secrecy means that the adversary cannot compute exactly the considered piece of data. In this survey, this property will simply be named *secrecy*, or when emphasis is needed, syntactic secrecy.
- Sometimes, one uses a stronger notion, *strong secrecy*, which means that the adversary cannot detect a change in the value of the secret (Abadi, 1999; Blanchet, 2004). In other words, the adversary has no information at all on the value of the secret.

The difference between syntactic secrecy and strong secrecy can be illustrated by a simple example: consider a piece of data for which the adversary knows half of the bits but not the other half. This piece of data is syntactically secret since the adversary cannot compute it entirely, but not strongly secret, since the adversary can see if one of the bits it knows changes. Syntactic secrecy cannot be used to express secrecy of data chosen among known constants. For instance, talking about syntactic secrecy of a boolean true or false does not make sense, because the adversary knows the constants true and false from the start. In this case, one has to use strong secrecy: the adversary must not be able to distinguish a protocol using the value true from the same protocol using the value false. These two notions are often equivalent (Cortier *et al.*, 2007), for atomic data (data that cannot be split into several pieces, such as nonces, which are random numbers chosen independently at each run of the protocol) and for probabilistic cryptographic primitives. Syntactic secrecy is a trace property, while strong secrecy is an equivalence property.

### Authentication

Authentication means that, if a participant $A$ runs the protocol apparently with a participant $B$, then $B$ runs the protocol apparently with $A$, and conversely. One often requires that $A$ and $B$ also share the same values of the parameters of the protocol.

Authentication is generally formalized by correspondence properties (Woo and Lam, 1993; Lowe, 1997), of the form: if $A$ executes a certain event $e_1$ (for instance, $A$ terminates the protocol with $B$), then $B$ has executed a certain event $e_2$ (for instance, $B$ started a session of the protocol with $A$). There exist several variants of these properties. For instance, one may require that each execution of $e_1$ corresponds to a distinct execution of $e_2$ (injective correspondence) or, on the contrary, that if $e_1$ has been executed, then $e_2$ has been executed at least once (non-injective correspondence). The events $e_1$ and $e_2$ may also include more or fewer parameters depending on the desired property. These properties are trace properties.

### Symbolic verification

Basically, to verify protocols in the symbolic model, one computes the set of terms (messages) that the adversary knows. If a message does not belong to this set, then this message is secret. The difficulty is that this set is infinite, for two reasons: the adversary can build terms of unbounded size, and the considered protocol can be executed any number of times. Several approaches can be considered to solve this problem:

- One can bound the size of messages and the number of executions of the protocols. In this case, the state space is finite, and one can apply standard model-checking techniques. This is the approach taken by FDR (Lowe, 1996) and by SATMC (Armando *et al.*, 2014), for instance.
- If we bound only the number of executions of the protocol, the state space is infinite, but under reasonable assumptions, one can show that the problem of security protocol verification is decidable: protocol insecurity is NP-complete (Rusinowitch and Turuani, 2003). Basically, the non-deterministic Turing machine guesses an attack and polynomially checks that it is actually an attack against the protocol. There exist practical tools that can verify protocols in this case, using for instance constraint solving as in Cl-AtSe (Turuani, 2006) or extensions of model checking as in OFMC (Basin *et al.*, 2005).
- When the number of executions of the protocol is not bounded, the problem is undecidable (Durgin *et al.*, 2004) for a reasonable model of protocols. Hence, there exists no automatic tool that always terminates and solves this problem. However, there are several approaches that can tackle an undecidable problem:
  - One can rely on help from the user. This is the approach taken for example by Isabelle (Paulson, 1998), which is an interactive theorem prover, Tamarin (Schmidt *et al.*, 2012), which just requires the user to give a few lemmas to help the tool, or Cryptyc (Gordon and Jeffrey, 2004), which relies on typing with type annotations.
  - One can have incomplete tools, which sometimes answer "I don't know" but succeed on many practical examples. For instance, one can use abstractions based on tree-automata to

represent the knowledge of the adversary (Monniaux, 2003; Boichut *et al.*, 2006).

– One can allow non-termination, as in Maude-NPA (Meadows, 1996; Escobar *et al.*, 2006).

The symbolic protocol verifier ProVerif represents protocols by Horn clauses, in the line of ideas by Weidenbach (1999): Horn clauses are first order logical formulas, of the form $F_1 \wedge \cdots \wedge F_n \Rightarrow F$, where $F_1, \ldots, F_n, F$ are facts. This representation introduces abstractions. It is still more precise than tree-automata because it keeps relational information on messages. However, using this approach, termination is not guaranteed in general.

Let us compare ProVerif with some other tools that verify protocol specifications in the symbolic model. AVISPA (Armando *et al.*, 2005) is a platform that offers four different protocol verification back-ends: SATMC (Armando *et al.*, 2014) for bounded attack depth (which implies bounded sessions and messages), Cl-AtSe (Turuani, 2006) and OFMC (Basin *et al.*, 2005; Mödersheim and Viganò, 2009) for bounded sessions, and TA4SP (Boichut *et al.*, 2006) for unbounded sessions. In contrast, ProVerif focuses only on the case of unbounded sessions, and the Horn-clause abstraction it uses is more precise than the tree-automata abstraction of TA4SP, as mentioned above. SATMC supports basic cryptographic primitives that can be defined by rewrite rules. Cl-AtSe additionally supports exclusive or, Diffie-Hellman exponentiation (including equations of the multiplicative group modulo $p$), and associative concatenation. OFMC supports cryptographic primitives defined by finite equational theories (theories under which every term has a finite equivalence class) and subterm convergent theories (theories generated by rewrite rules that are convergent, that is, terminating and confluent, and whose right-hand side is either a subterm of the left-hand side or a constant). However, in order to guarantee termination, it bounds the number of instantiations of variables. TA4SP handles algebraic properties of exponentiation and exclusive or. ProVerif supports cryptographic primitives defined by rewrite rules and by equations that satisfy the finite variant property (Comon-Lundh and Delaune, 2005), which excludes associativity. AVISPA focuses on trace properties, while ProVerif can also verify some equivalence properties.

Maude-NPA (Meadows, 1996; Escobar *et al.*, 2006) relies on narrowing in rewrite systems. It is fully automatic and supports an unbounded number of sessions, but in contrast to ProVerif, it does not make any abstraction. Hence, it is sound and complete, but may not terminate. It supports cryptographic primitives defined by convergent rewrite rules plus associativity and commutativity (Escobar *et al.*, 2007), as well as homomorphic encryption (Escobar *et al.*, 2011), while ProVerif does not support associativity nor homomorphic encryption. It initially focused on reachability properties and was recently extended to prove some equivalences (Santiago *et al.*, 2014), using the same idea as ProVerif (see §3).

Scyther (Cremers, 2008) is fully automatic, always terminates, and can provide three different results: verification for an unbounded number of sessions, attack, or verification for a bounded number of sessions. It supports only a fixed set of cryptographic primitives (symmetric and asymmetric encryption and signatures). It proves secrecy and authentication properties. A version named scyther-proof generates Isabelle proofs of security of the verified protocols (Meier *et al.*, 2010).

Tamarin (Schmidt *et al.*, 2012) verifies protocols for an unbounded number of sessions, but often relies on the user to provide some lemmas in order to guide the proof. It initially proved trace properties expressed in temporal first-order logic, and was recently extended to prove some equivalences (Basin *et al.*, 2015), using the same idea as ProVerif. It supports cryptographic primitives defined by subterm convergent equations, Diffie-Hellman exponentiation, bilinear pairings, and associative and commutative operators (Schmidt *et al.*, 2014). It also supports mutable state and loops; the lemmas provided by the user basically give loop invariants. Protocols in Tamarin are specified as multiset rewriting systems; Kremer and Künnemann (2014) wrote a translator from an extension of the applied pi calculus with state.

The rest of this survey focuses on ProVerif. We refer the reader to (Blanchet, 2012b) for a more complete survey of security protocol verification.

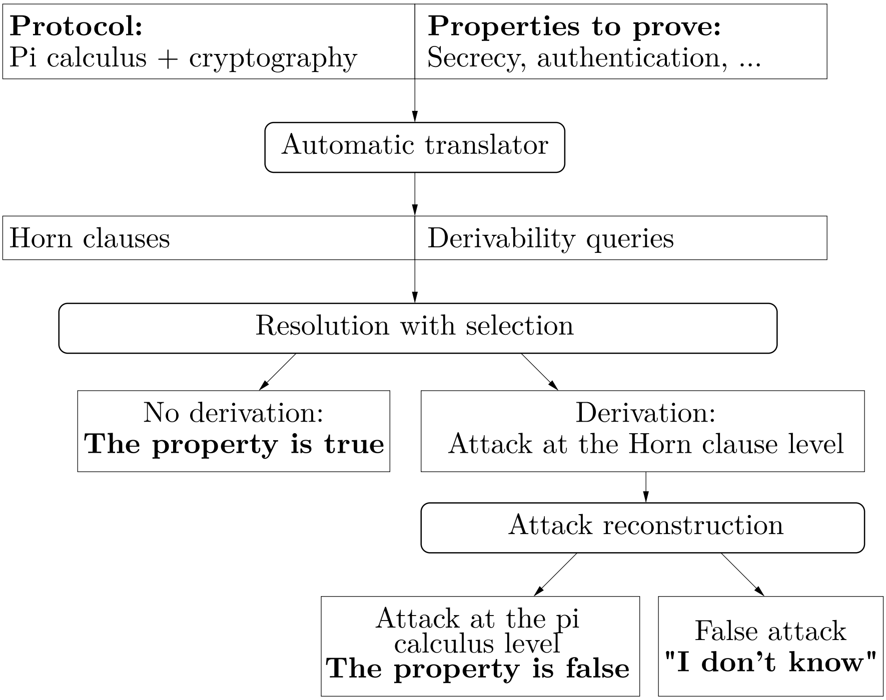


**Figure 1.1:** Structure of ProVerif

### Structure of ProVerif

The structure of ProVerif is represented in Figure 1.1. ProVerif takes as input a model of the protocol in an extension of the pi calculus with cryptography, similar to the applied pi calculus (Abadi and Fournet, 2001; Abadi *et al.*, 2016) and detailed in the next chapter. It supports a wide variety of cryptographic primitives, modeled by rewrite rules or by equations. ProVerif also takes as input the security properties that we want to prove. It can verify various security properties, including secrecy, authentication, and some observational equivalence properties. It automatically translates this information into an internal representation by Horn clauses: the protocol is translated into a set of Horn clauses, and the security properties to prove are translated into derivability queries on these clauses. ProVerif uses an algorithm based on resolution with free selection to determine whether a fact is derivable from the clauses. If the fact is *not* derivable, then the desired security property is proved. If the fact is derivable, then there may be an attack against the considered property: the derivation may correspond to an attack, but it may also correspond to a "false attack", because the Horn clause representation makes some abstractions. These abstractions are key to the verification of an unbounded number of sessions of protocols.

Chapter 2 presents the protocol specification language of ProVerif. Chapter 3 explains how ProVerif verifies the desired security properties. Chapter 4 relates the protocol specification language of ProVerif to the applied pi calculus (Abadi and Fournet, 2001; Abadi *et al.*, 2016). Finally, Chapter 5 summarizes some applications of ProVerif and Chapter 6 concludes.

## Comparison with previous surveys

Previous surveys on ProVerif (Blanchet, 2011; Blanchet, 2014) focus only on secrecy. The general protocol verification survey Blanchet (2012b) also outlines the verification of secrecy in ProVerif. Previous journal papers present individual features of the tool: secrecy (Abadi and Blanchet, 2005a), correspondences (Blanchet, 2009), and equivalences Blanchet *et al.* (2008). Our habilitation thesis (Blanchet, 2008b), in French, presents a general survey of ProVerif that includes secrecy, correspondences, and equivalences.

This survey is the first one to present all these features in English, in a common framework. Moreover, it includes features that never appeared in previous surveys: the extended destructors of (Cheval and Blanchet, 2013), the proof of equivalences using swapping (Blanchet and Smyth, 2016), as well as the link with the applied pi calculus (Chapter 4), which was never published before.

# References


Abadi, M. 1999. “Secrecy by Typing in Security Protocols”. *Journal of the ACM*. 46(5): 749–786.

Abadi, M. and B. Blanchet. 2005a. “Analyzing Security Protocols with Secrecy Types and Logic Programs”. *Journal of the ACM*. 52(1): 102–146.

Abadi, M. and B. Blanchet. 2005b. “Computer-Assisted Verification of a Protocol for Certified Email”. *Science of Computer Programming*. 58(1–2): 3–27. Special issue SAS’03.

Abadi, M., B. Blanchet, and C. Fournet. 2007. “Just Fast Keying in the Pi Calculus”. *ACM Transactions on Information and System Security (TISSEC)*. 10(3): 1–59.

Abadi, M., B. Blanchet, and C. Fournet. 2016. “The Applied Pi Calculus: Mobile Values, New Names, and Secure Communication”. Report arXiv:1609.03003v1. Available at http://arxiv.org/abs/1609.03003v1.

Abadi, M. and V. Cortier. 2006. “Deciding Knowledge in Security Protocols under Equational Theories”. *Theoretical Computer Science*. 367(1–2): 2–32.

Abadi, M. and C. Fournet. 2001. “Mobile Values, New Names, and Secure Communication”. In: *28th ACM Symposium on Principles of Programming Languages (POPL’01)*. London, UK: ACM. 104–115.

Abadi, M. and C. Fournet. 2004. "Private authentication". *Theoretical Computer Science*. 322(3): 427–476.

Abadi, M., N. Glew, B. Horne, and B. Pinkas. 2002. "Certified Email with a Light On-line Trusted Third Party: Design and Implementation". In: *11th International World Wide Web Conference*. Honolulu, Hawaii: ACM. 387–395.

Abadi, M. and A. D. Gordon. 1998. "A Bisimulation Method for Cryptographic Protocols". *Nordic Journal of Computing*. 5(4): 267–303.

Abadi, M. and A. D. Gordon. 1999. "A Calculus for Cryptographic Protocols: The Spi Calculus". *Information and Computation*. 148(1): 1–70. An extended version appeared as Digital Equipment Corporation Systems Research Center report No. 149, January 1998.

Abadi, M. and R. Needham. 1996. "Prudent Engineering Practice for Cryptographic Protocols". *IEEE Transactions on Software Engineering*. 22(1): 6–15.

Abadi, M. and P. Rogaway. 2002. "Reconciling Two Views of Cryptography (The Computational Soundness of Formal Encryption)". *Journal of Cryptology*. 15(2): 103–127.

Abdalla, M., P.-A. Fouque, and D. Pointcheval. 2005. "Password-Based Authenticated Key Exchange in the Three-Party Setting". In: *2005 International Workshop on Practice and Theory in Public Key Cryptography (PKC'05)*. Ed. by S. Vaudenay. Vol. 3386. *Lecture Notes in Computer Science*. Les Diablerets, Switzerland: Springer. 65–84.

Adrian, D., K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, J. A. Halderman, N. Heninger, D. Springall, E. Thomé, L. Valenta, B. VanderSloot, E. Wustrow, S. Zanella-Béguelin, and P. Zimmermann. 2015. "Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice". In: *22nd ACM Conference on Computer and Communications Security*.

Aiello, W., S. M. Bellovin, M. Blaze, R. Canetti, J. Ioannidis, K. Keromytis, and O. Reingold. 2004. "Just Fast Keying: Key Agreement in a Hostile Internet". *ACM Transactions on Information and System Security*. 7(2): 242–273.

Aizatulin, M., A. D. Gordon, and J. Jürjens. 2011. "Extracting and Verifying Cryptographic Models from C Protocol Code by Symbolic Execution". In: *18th ACM Conference on Computer and Communications Security (CCS'11)*. Chicago, IL, USA: ACM. 331–340.

Aizatulin, M., A. D. Gordon, and J. Jürjens. 2012. "Computational Verification of C Protocol Implementations by Symbolic Execution". In: *19th ACM Conference on Computer and Communications Security (CCS'12)*. Raleigh, NC, USA: ACM. 712–723.

Allamigeon, X. and B. Blanchet. 2005. "Reconstruction of Attacks against Cryptographic Protocols". In: *18th IEEE Computer Security Foundations Workshop (CSFW-18)*. Aix-en-Provence, France: IEEE. 140–154.

Almeida, J. B., M. Barbosa, G. Barthe, and F. Dupressoir. 2013. "Certified computer-aided cryptography: efficient provably secure machine code from high-level implementations". In: *ACM Conference on Computer and Communications Security (CCS'13)*. Berlin, Germany: ACM. 1217–1230.

Arapinis, M. and M. Duflot. 2007. "Bounding Messages for Free in Security Protocols". In: *27th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'07)*. Ed. by V. Arvind and S. Prasad. Vol. 4855. *Lecture Notes in Computer Science*. New Delhi, India: Springer. 376–387.

Arapinis, M., J. Liu, E. Ritter, and M. Ryan. 2014. "Stateful Applied Pi Calculus". In: *Principles of Security and Trust—Third International Conference*. Ed. by M. Abadi and S. Kremer. Vol. 8414. *Lecture Notes in Computer Science*. Springer. 22–41.

Arapinis, M., E. Ritter, and M. D. Ryan. 2011. "StatVerif: Verification of stateful processes". In: *24th Computer Security Foundations Symposium (CSF'11)*. IEEE. Cernay-la-Ville, France. 33–47.

Armando, A., D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. H. Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganó, and L. Vigneron. 2005. "The AVISPA tool for Automated Validation of Internet Security Protocols and Applications". In: *Computer Aided Verification, 17th International Conference, CAV 2005*. Ed. by K. Etessami and S. K. Rajamani. Vol. 3576. *Lecture Notes in Computer Science*. Edinburgh, Scotland: Springer. 281–285.

Armando, A., R. Carbone, and L. Compagna. 2014. "SATMC: a SAT-based Model Checker for Security-critical Systems". In: *Tools and Algorithms for the Construction and Analysis of Systems, 20th International Conference, TACAS 2014*. Ed. by E. Ábrahám and K. Havelund. Vol. 8413. *Lecture Notes in Computer Science*. Grenoble, France: Springer. 31–45. DOI: 10.1007/978-3-642-54862-8_3.

Avalle, M., A. Pironti, R. Sisto, and D. Pozza. 2011. "The JavaSPI Framework for Security Protocol Implementation". In: *International Conference on Availability, Reliability and Security (ARES'11)*. IEEE. 746–751.

Bachmair, L. and H. Ganzinger. 2001. "Resolution Theorem Proving". In: *Handbook of Automated Reasoning*. Ed. by A. Robinson and A. Voronkov. Vol. 1. North Holland. Chap. 2. 19–100.

Backes, M., F. Bendun, M. Maffei, E. Mohammadi, and K. Pecina. 2015. "Symbolic Malleable Zero-Knowledge Proofs". In: *28th IEEE Computer Security Foundations Symposium (CSF'15)*. Verona, Italy: IEEE. 412–480.

Backes, M., C. Hritcu, and M. Maffei. 2008a. "Automated Verification of Remote Electronic Voting Protocols in the Applied Pi-calculus". In: *21st IEEE Computer Security Foundations Symposium (CSF'08)*. Pittsburgh, PA: IEEE Computer Society. 195–209.

Backes, M., M. Maffei, and D. Unruh. 2008b. "Zero-Knowledge in the Applied Pi-calculus and Automated Verification of the Direct Anonymous Attestation Protocol". In: *29th IEEE Symposium on Security and Privacy*. Technical report version available at http://eprint.iacr.org/2007/289. IEEE. Oakland, CA. 202–215.

Backes, M., E. Mohammadi, and T. Ruffing. 2014. "Computational Soundness Results for ProVerif: Bridging the Gap from Trace Properties to Uniformity". In: *Principles of Security and Trust (POST'14)*. Ed. by M. Abadi and S. Kremer. Vol. 8414. *Lecture Notes in Computer Science*. Grenoble, France: Springer. 42–62.

Bansal, C., K. Bhargavan, A. Delignat-Lavaud, and S. Maffeis. 2013. "Keys to the Cloud: Formal Analysis and Concrete Attacks on Encrypted Web Storage". In: *Principles of Security and Trust (POST 2013)*. Ed. by D. Basin and J. Mitchell. Vol. 7796. *Lecture Notes in Computer Science*. Rome, Italy: Springer. 126–146.

Bansal, C., K. Bhargavan, and S. Maffeis. 2012. "Discovering Concrete Attacks on Website Authorization by Formal Analysis". In: *25th IEEE Computer Security Foundations Symposium (CSF'12)*. IEEE. Cambridge, MA, USA. 247–262.

Barthe, G., F. Dupressoir, P.-A. Fouque, B. Grégoire, M. Tibouchi, and J.-C. Zapalowicz. 2014a. "Making RSA-PSS Provably Secure against Non-random Faults". In: *Cryptographic Hardware and Embedded Systems (CHES'14)*. Ed. by L. Batina and M. Robshaw. Vol. 8731. *Lecture Notes in Computer Science*. Busan, South Korea: Springer. 206–222.

Barthe, G., F. Dupressoir, B. Grégoire, C. Kunz, B. Schmidt, and P.-Y. Strub. 2014b. "EasyCrypt: A Tutorial". In: *Foundations of Security Analysis and Design VII*. Ed. by A. Aldini, J. Lopez, and F. Martinelli. Vol. 8604. *Lecture Notes in Computer Science*. Springer. 146–166.

Barthe, G., B. Grégoire, S. Heraud, and S. Z. Béguelin. 2011. "Computer-Aided Security Proofs for the Working Cryptographer". In: *Advances in Cryptology – CRYPTO 2011*. Ed. by P. Rogaway. Vol. 6841. *Lecture Notes in Computer Science*. Santa Barbara, CA, USA: Springer. 71–90.

Barthe, G., B. Grégoire, and S. Zanella. 2009. "Formal Certification of Code-Based Cryptographic Proofs". In: *36th ACM SIGPLAN - SIGACT Symposium on Principles of Programming Languages (POPL'09)*. Savannah, Georgia: ACM. 90–101.

Basin, D., J. Dreier, and R. Casse. 2015. “Automated Symbolic Proofs of Observational Equivalence”. In: *22nd ACM Conference on Computer and Communications Security (CCS'15)*. Denver, CO: ACM. 1144–1155.

Basin, D., S. Mödersheim, and L. Viganò. 2005. “OFMC: A symbolic model checker for security protocols”. *International Journal of Information Security*. 4(3): 181–208.

Baudet, M. 2007. “Sécurité des protocoles cryptographiques: aspects logiques et calculatoires”. *PhD thesis*. Ecole Normale Supérieure de Cachan.

Béguelin, S. Z., B. Grégoire, G. Barthe, and F. Olmedo. 2009. “Formally Certifying the Security of Digital Signature Schemes”. In: *30th IEEE Symposium on Security and Privacy, S&P 2009*. Oakland, CA: IEEE. 237–250.

Bellovin, S. M. and M. Merritt. 1992. “Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks”. In: *1992 IEEE Computer Society Symposium on Research in Security and Privacy*. 72–84.

Bellovin, S. M. and M. Merritt. 1993. “Augmented Encrypted Key Exchange: a Password-Based Protocol Secure Against Dictionary Attacks and Password File Compromise”. In: *First ACM Conference on Computer and Communications Security*. 244–250.

Bengtson, J., K. Bhargavan, C. Fournet, A. Gordon, and S. Maffeis. 2011. “Refinement Types for Secure Implementations”. *ACM Transactions on Programming Languages and Systems*. 33(2).

Beurdouche, B., K. Bhargavan, A. Delignat-Lavaud, C. Fournet, M. Kohlweiss, A. Pironti, P.-Y. Strub, and J. K. Zinzindohoue. 2015. “A Messy State of the Union: Taming the Composite State Machines of TLS”. In: *IEEE Symposium on Security & Privacy 2015 (Oakland'15)*. IEEE.

Bhargavan, K., R. Corin, and C. Fournet. 2007. “Crypto-Verifying Protocol Implementations in ML”. http://doc.utwente.nl/64107/1/fs2cv.pdf.

Bhargavan, K., R. Corin, C. Fournet, and E. Zălinescu. 2008. "Cryptographically Verified Implementations for TLS". In: *15th ACM Conference on Computer and Communications Security (CCS'08)*. ACM. 459–468.

Bhargavan, K., C. Fournet, and A. Gordon. 2004. "Verifying Policy-Based Security for Web Services". In: *ACM Conference on Computer and Communications Security (CCS'04)*. Washington DC: ACM. 268–277.

Bhargavan, K., C. Fournet, and A. Gordon. 2010. "Modular Verification of Security Protocol Code by Typing". In: *ACM Symposium on Principles of Programming Languages (POPL'10)*. Madrid, Spain: ACM. 445–456.

Bhargavan, K., C. Fournet, A. Gordon, and S. Tse. 2006. "Verified interoperable implementations of security protocols". In: *19th IEEE Computer Security Foundations Workshop (CSFW'06)*. Venice, Italy: IEEE Computer Society. 139–152.

Bhargavan, K., C. Fournet, M. Kohlweiss, A. Pironti, and P.-Y. Strub. 2013. "Implementing TLS with Verified Cryptographic Security". In: *IEEE Symposium on Security & Privacy*. 445–462.

Blanchet, B. 2004. "Automatic Proof of Strong Secrecy for Security Protocols". In: *IEEE Symposium on Security and Privacy*. Oakland, California. 86–100.

Blanchet, B. 2008a. "A Computationally Sound Mechanized Prover for Security Protocols". *IEEE Transactions on Dependable and Secure Computing*. 5(4): 193–207.

Blanchet, B. 2008b. "Vérification automatique de protocoles cryptographiques : modèle formel et modèle calculatoire". *Mémoire d'habilitation à diriger des recherches*. Université Paris-Dauphine.

Blanchet, B. 2009. "Automatic Verification of Correspondences for Security Protocols". *Journal of Computer Security*. 17(4): 363–434.

Blanchet, B. 2011. "Using Horn Clauses for Analyzing Security Protocols". In: *Formal Models and Techniques for Analyzing Security Protocols*. Ed. by V. Cortier and S. Kremer. Vol. 5. *Cryptology and Information Security Series*. IOS Press. 86–111.

Blanchet, B. 2012a. "Mechanizing Game-Based Proofs of Security Protocols". In: *Software Safety and Security - Tools for Analysis and Verification*. Ed. by T. Nipkow, O. Grumberg, and B. Hauptmann. Vol. 33. *NATO Science for Peace and Security Series – D: Information and Communication Security*. Proceedings of the 2011 MOD summer school. IOS Press. 1–25.

Blanchet, B. 2012b. "Security Protocol Verification: Symbolic and Computational Models". In: *First Conference on Principles of Security and Trust (POST'12)*. Ed. by P. Degano and J. Guttman. Vol. 7215. *Lecture Notes in Computer Science*. Tallinn, Estonia: Springer. 3–29.

Blanchet, B. 2014. "Automatic Verification of Security Protocols in the Symbolic Model: the Verifier ProVerif". In: *Foundations of Security Analysis and Design VII, FOSAD Tutorial Lectures*. Ed. by A. Aldini, J. Lopez, and F. Martinelli. Vol. 8604. *Lecture Notes in Computer Science*. Springer. 54–87.

Blanchet, B., M. Abadi, and C. Fournet. 2008. "Automated Verification of Selected Equivalences for Security Protocols". *Journal of Logic and Algebraic Programming*. 75(1): 3–51.

Blanchet, B. and A. Chaudhuri. 2008. "Automated Formal Analysis of a Protocol for Secure File Sharing on Untrusted Storage". In: *IEEE Symposium on Security and Privacy*. IEEE. Oakland, CA. 417–431.

Blanchet, B. and A. Podelski. 2005. "Verification of Cryptographic Protocols: Tagging Enforces Termination". *Theoretical Computer Science*. 333(1-2): 67–90. Special issue FoSSaCS'03.

Blanchet, B. and B. Smyth. 2016. "Automated reasoning for equivalences in the applied pi calculus with barriers". In: *29th IEEE Computer Security Foundations Symposium (CSF'16)*. Lisboa, Portugal: IEEE. 310–324.

Blanchet, B., B. Smyth, and V. Cheval. 2016. "ProVerif 1.94pl1: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial". Available at http://proverif.inria.fr/manual.pdf.

Boichut, Y., N. Kosmatov, and L. Vigneron. 2006. "Validation of Prouvé protocols using the automatic tool TA4SP". In: *Third Taiwanese-French Conference on Information Technology (TFIT 2006)*. Nancy, France. 467–480.

Bruni, A., S. Mödersheim, F. Nielson, and H. R. Nielson. 2015. "Set-Pi: Set Membership Pi-Calculus". In: *28th IEEE Computer Security Foundations Symposium (CSF'15)*. Verona, Italy: IEEE. 185–198.

Cadé, D. and B. Blanchet. 2013. "From Computationally-Proved Protocol Specifications to Implementations and Application to SSH". *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*. 4(1): 4–31.

Cadé, D. and B. Blanchet. 2015. "Proved Generation of Implementations from Computationally Secure Protocol Specifications". *Journal of Computer Security*. 23(3): 331–402.

Canetti, R. and J. Herzog. 2006. "Universally Composable Symbolic Analysis of Mutual Authentication and Key Exchange Protocols". In: *Proceedings, Theory of Cryptography Conference (TCC'06)*. Ed. by S. Halevi and T. Rabin. Vol. 3876. *Lecture Notes in Computer Science*. Extended version available at http://eprint.iacr.org/2004/334. New York, NY: Springer. 380–403.

Chadha, R., S. Ciobâca, and S. Kremer. 2012. "Automated Verification of Equivalence Properties of Cryptographic Protocols". In: *21st European Symposium on Programming (ESOP'12)*. Vol. 7211. *Lecture Notes in Computer Science*. Springer. 108–127.

Chaki, S. and A. Datta. 2009. "ASPIER: An Automated Framework for Verifying Security Protocol Implementations". In: *22nd IEEE Computer Security Foundations Symposium (CSF'09)*. Port Jefferson, NY, USA. 172–185.

Cheval, V. and B. Blanchet. 2013. "Proving More Observational Equivalences with ProVerif". In: *2nd Conference on Principles of Security and Trust (POST 2013)*. Ed. by D. Basin and J. Mitchell. Vol. 7796. *Lecture Notes in Computer Science*. Rome, Italy: Springer. 226–246.

Cheval, V., H. Comon-Lundh, and S. Delaune. 2011. "Trace Equivalence Decision: Negative Tests and Non-determinism". In: *Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS'11)*. Chicago, Illinois, USA: ACM. 321–330.

Chevalier, Y., R. Küsters, M. Rusinowitch, and M. Turuani. 2003. "Deciding the Security of Protocols with Diffie-Hellman Exponentiation and Products in Exponents". In: *FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science, 23rd Conference*. Ed. by P. K. Pandya and J. Radhakrishnan. Vol. 2914. *Lecture Notes in Computer Science*. Mumbai, India: Springer. 124–135.

Chevalier, Y., R. Küsters, M. Rusinowitch, and M. Turuani. 2005. "An NP decision procedure for protocol insecurity with XOR". *Theoretical Computer Science*. 338(1–3): 247–274.

Chothia, T., B. Smyth, and C. Staite. 2015. "Automatically Checking Commitment Protocols in ProVerif without False Attacks". In: *Principles of Security and Trust, 4th International Conference, POST 2015*. Ed. by R. Focardi and A. Myers. Vol. 9036. *Lecture Notes in Computer Science*. London, UK: Springer. 137–155.

Chrétien, R., V. Cortier, and S. Delaune. 2015a. "Decidability of trace equivalence for protocols with nonces". In: *28th IEEE Computer Security Foundations Symposium (CSF'15)*. Verona, Italy: IEEE Computer Society. 170–184. DOI: 10.1109/CSF.2015.19.

Chrétien, R., V. Cortier, and S. Delaune. 2015b. "From security protocols to pushdown automata". *ACM Transactions on Computational Logic*. 17(1:3). DOI: 10.1145/2811262.

Ciobâcă, Ş. 2011. "Automated Verification of Security Protocols with Applications to Electronic Voting". *PhD thesis*. ENS Cachan.

Cohen, E. 2002. "Proving Protocols Safe from Guessing". In: *Foundations of Computer Security*. Copenhagen, Denmark.

Comon-Lundh, H. and V. Cortier. 2003. "New Decidability Results for Fragments of First-Order Logic and Application to Cryptographic Protocols". In: *14th Int. Conf. Rewriting Techniques and Applications (RTA'2003)*. Ed. by R. Nieuwenhuis. Vol. 2706. *Lecture Notes in Computer Science*. Valencia, Spain: Springer. 148–164.

Comon-Lundh, H. and S. Delaune. 2005. "The finite variant property: How to get rid of some algebraic properties". In: *Proceedings of the 16th International Conference on Rewriting Techniques and Applications (RTA'05)*. Ed. by J. Giesl. Vol. 3467. *Lecture Notes in Computer Science*. Nara, Japan: Springer. 294–307.

Comon-Lundh, H. and V. Shmatikov. 2003. "Intruder deductions, constraint solving and insecurity decision in presence of exclusive or". In: *Symposium on Logic in Computer Science (LICS'03)*. Ottawa, Canada: IEEE Computer Society. 271–280.

Corin, R., J. M. Doumen, and S. Etalle. 2004. "Analysing Password Protocol Security Against Off-line Dictionary Attacks". In: *2nd Int. Workshop on Security Issues with Petri Nets and other Computational Models (WISP). Electronic Notes in Theoretical Computer Science.*

Corin, R., S. Malladi, J. Alves-Foss, and S. Etalle. 2003. "Guess What? Here is a New Tool that Finds some New Guessing Attacks". In: *Workshop on Issues in the Theory of Security (WITS'03)*. Ed. by R. Gorrieri. Warsaw, Poland.

Cortier, V., H. Hördegen, and B. Warinschi. 2006. "Explicit Randomness is not Necessary when Modeling Probabilistic Encryption". In: *Workshop on Information and Computer Security (ICS 2006)*. Ed. by C. Dima, M. Minea, and F. Tiplea. Vol. 186. *Electronic Notes in Theoretical Computer Science*. Timisoara, Romania: Elsevier. 49–65.

Cortier, V., S. Kremer, and B. Warinschi. 2011. "A Survey of Symbolic Methods in Computational Analysis of Cryptographic Systems". *Journal of Automated Reasoning*. 46(3-4): 225–259.

Cortier, V., M. Rusinowitch, and E. Zălinescu. 2007. "Relating two standard notions of secrecy". *Logical Methods in Computer Science*. 3(3).

Cortier, V. and C. Wiedling. 2012. "A formal analysis of the Norwegian E-voting protocol". In: *Proceedings of the 1st International Conference on Principles of Security and Trust (POST'12)*. Ed. by P. Degano and J. D. Guttman. Vol. 7215. *Lecture Notes in Computer Science*. Tallinn, Estonia: Springer. 109–128.

Cremers, C. J. 2008. "Unbounded verification, falsification, and characterization of security protocols by pattern refinement". In: *15th ACM conference on Computer and Communications Security (CCS'08)*. Alexandria, Virginia, USA: ACM. 119–128.

Delaune, S. and F. Jacquemard. 2004. "A Theory of Dictionary Attacks and its Complexity". In: *17th IEEE Computer Security Foundations Workshop*. Pacific Grove, CA: IEEE. 2–15.

Delaune, S., S. Kremer, and M. D. Ryan. 2009. "Verifying Privacy-type Properties of Electronic Voting Protocols". *Journal of Computer Security*. 17(4): 435–487.

Delaune, S., S. Kremer, M. D. Ryan, and G. Steel. 2011. "Formal analysis of protocols based on TPM state registers". In: *Proceedings of the 24th IEEE Computer Security Foundations Symposium (CSF'11)*. Cernay-la-Ville, France: IEEE Computer Society. 66–82.

Delaune, S., M. Ryan, and B. Smyth. 2008. "Automatic verification of privacy properties in the applied pi calculus". In: *Second Joint iTrust and PST Conferences on Privacy, Trust Management and Security (IFIPTM'08)*. Ed. by Y. Karabulut, J. Mitchell, P. Herrmann, and C. D. Jensen. Vol. 263. *IFIP Advances in Information and Communication Technology*. Trondheim, Norway: Springer. 263–278.

Denning, D. E. and G. M. Sacco. 1981. "Timestamps in Key Distribution Protocols". *Communications of the ACM*. 24(8): 533–536.

Diffie, W. and M. Hellman. 1976. "New Directions in Cryptography". *IEEE Transactions on Information Theory*. IT-22(6): 644–654.

Dolev, D. and A. C. Yao. 1983. "On the Security of Public Key Protocols". *IEEE Transactions on Information Theory*. IT-29(12): 198–208.

Dreier, J., P. Lafourcade, and Y. Lakhnech. 2013. "Formal Verification of e-Auction Protocols". In: *Principles of Security and Trust (POST'13)*. Ed. by D. Basin and J. Mitchell. Vol. 7796. *Lecture Notes in Computer Science*. Rome, Italy: Springer. 247–266.

Drielsma, P. H., S. Mödersheim, and L. Viganò. 2005. "A Formalization of Off-line Guessing for Security Protocol Analysis". In: *Logic for Programming, Artificial Intelligence, and Reasoning: 11th International Conference, LPAR 2004*. Ed. by F. Baader and A. Voronkov. Vol. 3452. *Lecture Notes in Computer Science*. Montevideo, Uruguay: Springer. 363–379.

Dupressoir, F., A. D. Gordon, J. Jürjens, and D. A. Naumann. 2011. "Guiding a General-Purpose C Verifier to Prove Cryptographic Protocols". In: *24th IEEE Symposium on Computer Security Foundations (CSF'11)*. Paris, France: IEEE Computer Society. 3–17.

Durgin, N., P. Lincoln, J. C. Mitchell, and A. Scedrov. 2004. "Multiset Rewriting and the Complexity of Bounded Security Protocols". *Journal of Computer Security*. 12(2): 247–311.

Escobar, S., J. Hendrix, C. Meadows, and J. Meseguer. 2007. "Diffie-Hellman cryptographic reasoning in the Maude-NRL protocol analyzer". In: *Proc. 2nd International Workshop on Security and Rewriting Techniques (SecReT 2007)*.

Escobar, S., D. Kapur, C. Lynch, C. Meadows, J. Meseguer, P. Narendran, and R. Sasse. 2011. "Protocol analysis in Maude-NPA using unification modulo homomorphic encryption". In: *13th international ACM SIGPLAN symposium on Principles and practices of declarative programming (PPDP'11)*. Odense, Denmark: ACM. 65–76.

Escobar, S., C. Meadows, and J. Meseguer. 2006. "A rewriting-based inference system for the NRL Protocol Analyzer and its meta-logical properties". *Theoretical Computer Science*. 367(1-2): 162–202.

Fournet, C. and M. Kohlweiss. 2011. "Modular Cryptographic Verification by Typing". In: *7th Workshop on Formal and Computational Cryptography (FCC'11)*. Paris, France.

Godskesen, J. C. 2006. "Formal Verification of the ARAN Protocol Using the Applied Pi-calculus". In: *Sixth International IFIP WG 1.7 Workshop on Issues in the Theory of Security (WITS'06)*. Vienna, Austria. 99–113.

Gordon, A. and A. Jeffrey. 2004. "Types and Effects for Asymmetric Cryptographic Protocols". *Journal of Computer Security*. 12(3/4): 435–484.

Goubault-Larrecq, J. 2005. "Deciding $\mathcal{H}_1$ by resolution". *Information Processing Letters*. 95(3): 401–408.

Goubault-Larrecq, J. and F. Parrennes. 2005. "Cryptographic Protocol Analysis on Real C Code". In: *6th International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI'05)*. Ed. by R. Cousot. Vol. 3385. *Lecture Notes in Computer Science*. Paris, France: Springer. 363–379.

Heather, J., G. Lowe, and S. Schneider. 2000. "How to Prevent Type Flaw Attacks on Security Protocols". In: *13th IEEE Computer Security Foundations Workshop (CSFW-13)*. Cambridge, England. 255–268.

Hüttel, H. 2003. "Deciding Framed Bisimilarity". *Electronic Notes in Theoretical Computer Science*. 68(6): 1–20. Special issue Infinity'02: 4th International Workshop on Verification of Infinite-State Systems.

Kallahalla, M., E. Riedel, R. Swaminathan, Q. Wang, and K. Fu. 2003. "Plutus: Scalable secure file sharing on untrusted storage". In: *2nd Conference on File and Storage Technologies (FAST'03)*. San Francisco, CA: Usenix. 29–42.

Khurana, H. and H.-S. Hahm. 2006. "Certified Mailing Lists". In: *ACM Symposium on Communication, Information, Computer and Communication Security (ASIACCS'06)*. Taipei, Taiwan: ACM. 46–58.

Kowalski, R. 1974. "Predicate Logic as Programming Language". In: *Proceedings IFIP Congress*. Stockholm: North Holland. 569–574.

Kremer, S. and R. Künnemann. 2014. "Automated Analysis of Security Protocols with Global State". In: *35th IEEE Symposium on Security and Privacy (S&P'14)*. San Jose, CA, USA: IEEE Computer Society.

Kremer, S. and M. D. Ryan. 2005. "Analysis of an Electronic Voting Protocol in the Applied Pi Calculus". In: *Programming Languages and Systems: 14th European Symposium on Programming, ESOP 2005*. Ed. by M. Sagiv. Vol. 3444. *Lecture Notes in Computer Science*. Edimbourg, UK: Springer. 186–200.

Küsters, R. and T. Truderung. 2008. "Reducing protocol analysis with XOR to the XOR-free case in the Horn theory based approach". In: *15th ACM conference on Computer and communications security (CCS'08)*. Alexandria, Virginia, USA: ACM. 129–138.

Küsters, R. and T. Truderung. 2009. "Using ProVerif to Analyze Protocols with Diffie-Hellman Exponentiation". In: *22nd IEEE Computer Security Foundations Symposium (CSF'09)*. Port Jefferson, New York, USA: IEEE. 157–171.

Lowe, G. 1996. "Breaking and Fixing the Needham-Schroeder Public-Key Protocol using FDR". In: *Tools and Algorithms for the Construction and Analysis of Systems*. Vol. 1055. *Lecture Notes in Computer Science*. Springer. 147–166.

Lowe, G. 1997. "A Hierarchy of Authentication Specifications". In: *10th Computer Security Foundations Workshop (CSFW '97)*. IEEE Computer Society. Rockport, Massachusetts. 31–43.

Lowe, G. 2002. "Analyzing Protocols Subject to Guessing Attacks". In: *Workshop on Issues in the Theory of Security (WITS'02)*. Portland, Oregon.

Lux, K. D., M. J. May, N. L. Bhattad, and C. A. Gunter. 2005. "WSEmail: Secure Internet Messaging Based on Web Services". In: *International Conference on Web Services (ICWS'05)*. Orlando, Florida: IEEE Computer Society. 75–82.

Meadows, C. A. 1996. "The NRL Protocol Analyzer: An Overview". *Journal of Logic Programming*. 26(2): 113–131.

Meadows, C. and P. Narendran. 2002. "A Unification Algorithm for the Group Diffie-Hellman Protocol". In: *Workshop on Issues in the Theory of Security (WITS'02)*. Portland, Oregon.

Meier, S., C. Cremers, and D. Basin. 2010. "Strong Invariants for the Efficient Construction of Machine-Checked Protocol Security Proofs". In: *23rd IEEE Computer Security Foundations Symposium (CSF'10)*. Edinburgh, UK: IEEE. 231–245.

Milicia, G. 2002. "$\chi$-Spaces: Programming Security Protocols". In: *14th Nordic Workshop on Programming Theory (NWPT'02)*. Tallinn, Estonia.

Millen, J. 1999. "A Necessarily Parallel Attack". In: *Workshop on Formal Methods and Security Protocols (FMSP'99)*. Trento, Italy.

Milner, R., J. Parrow, and D. Walker. 1992. "A Calculus of Mobile Processes, parts I and II". *Information and Computation*. 100(Sept.): 1–40 and 41–77.

Mödersheim, S. 2010. “Abstraction by Set-Membership: Verifying Security Protocols and Web Services with Databases”. In: *17th ACM Conference on Computer and Communications Security (CCS 2010)*. ACM. Chicago, IL, USA. 351–360.

Mödersheim, S. and L. Viganò. 2009. “The Open-source Fixed-point Model Checker for Symbolic Analysis of Security Protocols”. In: *Foundations of Security Analysis and Design V, FOSAD 2007 / 2008 / 2009 Tutorial Lectures*. Ed. by A. Aldini, G. Barthe, and R. Gorrieri. Vol. 5705. *Lecture Notes in Computer Science*. Springer. 166–194.

Monniaux, D. 2003. “Abstracting Cryptographic Protocols with Tree Automata”. *Science of Computer Programming*. 47(2–3): 177–202.

Mukhamedov, A., A. D. Gordon, and M. Ryan. 2013. “Towards a Verified Reference Implementation of a Trusted Platform Module”. In: *Security Protocols XVII*. Ed. by B. Christianson, J. A. Malcolm, V. Matyáš, and M. Roe. Vol. 7028. *Lecture Notes in Computer Science*. Springer. 69–81.

Needham, R. M. and M. D. Schroeder. 1978. “Using Encryption for Authentication in Large Networks of Computers”. *Communications of the ACM*. 21(12): 993–999.

O’Shea, N. 2008. “Using Elyjah to Analyse Java Implementations of Cryptographic Protocols”. In: *Joint Workshop on Foundations of Computer Security, Automated Reasoning for Security Protocol Analysis and Issues in the Theory of Security (FCS-ARSPA-WITS’08)*. Pittsburgh, PA, USA.

Pankova, A. and P. Laud. 2012. “Symbolic Analysis of Cryptographic Protocols Containing Bilinear Pairings”. In: *25th IEEE Computer Security Foundations Symposium (CSF’12)*. Cambridge, MA: IEEE. 63–77.

Paulson, L. C. 1998. “The Inductive Approach to Verifying Cryptographic Protocols”. *Journal of Computer Security*. 6(1–2): 85–128.

Pironti, A. and R. Sisto. 2010. “Provably Correct Java Implementations of Spi Calculus Security Protocols Specifications”. *Computers and Security*. 29(3): 302–314.

Pottier, F. 2002. “A Simple View of Type-Secure Information Flow in the $\pi$-Calculus”. In: *15th IEEE Computer Security Foundations Workshop*. Cape Breton, Nova Scotia. 320–330.

Pottier, F. and V. Simonet. 2002. “Information Flow Inference for ML”. In: *29th ACM Symposium on Principles of Programming Languages (POPL’02)*. Portland, Oregon. 319–330.

Pozza, D., R. Sisto, and L. Durante. 2004. “Spi2Java: Automatic cryptographic protocol Java code generation from spi calculus”. In: *18th International Conference on Advanced Information Networking and Applications (AINA’04)*. Vol. 1. Fukuoka, Japan: IEEE Computer Society. 400–405.

Ramanujam, R. and S. Suresh. 2003. “Tagging Makes Secrecy Decidable with Unbounded Nonces as Well”. In: *FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science*. Ed. by P. Pandya and J. Radhakrishnan. Vol. 2914. *Lecture Notes in Computer Science*. Mumbai, India: Springer. 363–374.

Rusinowitch, M. and M. Turuani. 2003. “Protocol Insecurity with Finite Number of Sessions is NP-complete”. *Theoretical Computer Science*. 299(1–3): 451–475.

Santiago, S., S. Escobar, C. Meadows, and J. Meseguer. 2014. “A Formal Definition of Protocol Indistinguishability and Its Verification Using Maude-NPA”. In: *Security and Trust Management (STM’14)*. Ed. by S. Mauw and C. D. Jensen. Vol. 8743. *Lecture Notes in Computer Science*. Wroclaw, Poland: Springer. 162–177.

Schmidt, B., S. Meier, C. Cremers, and D. Basin. 2012. “Automated Analysis of Diffie-Hellman Protocols and Advanced Security Properties”. In: *25th IEEE Computer Security Foundations Symposium (CSF’12)*. Cambridge, MA, USA: IEEE Computer Society. 78–94.

Schmidt, B., R. Sasse, C. Cremers, and D. Basin. 2014. “Automated Verification of Group Key Agreement Protocols”. In: *2014 IEEE Symposium on Security and Privacy*. San Jose, CA: IEEE. 179–194.

Smyth, B., M. D. Ryan, and L. Chen. 2015. “Formal analysis of privacy in Direct Anonymous Attestation schemes”. *Science of Computer Programming*. 111(2): 300–317.

Song, D., A. Perrig, and D. Phan. 2001. "AGVI—Automatic Generation, Verification, and Implementation of Security Protocols". In: *Computer Aided Verification (CAV'01)*. Ed. by G. Berry, H. Comon, and A. Finkel. Vol. 2102. *Lecture Notes in Computer Science*. Paris, France: Springer. 241–245.

Swamy, N., J. Chen, C. Fournet, P.-Y. Strub, K. Bhargavan, and J. Yang. 2011. "Secure Distributed Programming with Value-dependent Types". In: *16th International Conference on Functional Programming (ICFP 2011)*. Tokyo, Japan: ACM. 266–278.

Tiu, A. and J. Dawson. 2010. "Automating Open Bisimulation Checking for the Spi Calculus". In: *23rd IEEE Computer Security Foundations Symposium (CSF'10)*. Edinburgh, UK: IEEE. 307–321.

Turuani, M. 2006. "The CL-Atse Protocol Analyser". In: *Term Rewriting and Applications, 17th International Conference, RTA 2006*. Ed. by F. Pfenning. Vol. 4098. *Lecture Notes in Computer Science*. Seattle, WA: Springer. 277–286.

Weidenbach, C. 1999. "Towards an Automatic Analysis of Security Protocols in First-Order Logic". In: *16th International Conference on Automated Deduction (CADE-16)*. Ed. by H. Ganzinger. Vol. 1632. *Lecture Notes in Artificial Intelligence*. Trento, Italy: Springer. 314–328.

Woo, T. Y. C. and S. S. Lam. 1993. "A Semantic Model for Authentication Protocols". In: *IEEE Symposium on Research in Security and Privacy*. Oakland, California. 178–194.