Formal Verification of Security Protocols: 25 Years of ProVerif

Stéphanie Delaune
Univ Rennes, CNRS, IRISA, France

Abstract

Cryptographic protocols are essential to secure communication but remain difficult to design correctly, motivating the need for rigorous verification methods. This paper provides a brief overview of symbolic verification techniques, focusing on the evolution and impact of the tool ProVerif over the past 25 years. We discuss its core principles, key extensions for richer security properties, and recent work on handling algebraic theories such as exclusive-or (XOR). We conclude by highlighting ongoing challenges, including usability and interoperability between verification approaches.

2012 ACM Subject Classification Security and privacy → Logic and verification

Keywords and phrases Security protocols, Formal verification

Digital Object Identifier 10.4230/LIPIcs.LICS.2026.2

Category Invited Paper

Funding This work received funding from the France 2030 program managed by the French National Research Agency under grant agreement No. ANR-22-PECY-0006.

1 Introduction

Cryptographic protocols are at the core of security in modern digital communications. They underpin everyday systems such as secure web browsing, messaging applications, and authentication infrastructures. Despite their apparent simplicity, these protocols are notoriously difficult to design correctly. Subtle flaws often escape detection by traditional testing-based approaches, yet may lead to severe security breaches when deployed at scale.

This situation has motivated the development of formal verification techniques for cryptographic protocols. Rather than relying on empirical testing, these methods aim to provide mathematically rigorous guarantees that security properties such as confidentiality, or authentication hold against all possible adversarial behaviors. In practice, however, manual reasoning is error-prone and does not scale to realistic protocols. This has led to an increasing need for automated or at least mechanised verification techniques [6].

Two main paradigms coexist for modeling cryptographic protocols: the *symbolic* and the *computational* approaches. In the computational model, cryptography is grounded in complexity-theoretic assumptions, while in the symbolic approach, cryptographic messages are modeled as first-order terms, possibly along with an equational theory representing attacker capabilities. This idea, originally proposed in [33], has been refined over the years, giving rise to a variety of symbolic models. These models capture a wide range of attacker capabilities and enable the automated verification of protocols. Although abstract, the symbolic approach has proven remarkably successful in practice. It enables a high degree of automation and supports the analysis of complex, concurrent scenarios with potentially unbounded numbers of protocol sessions.

Among the most widely used symbolic verification tools are ProVerif [10] and Tamarin [43], which have proven effective for the analysis of real-world protocols. These tools have been successfully applied to protocols that underpin widely deployed systems and critical infrastructures, including the TLS 1.3 protocol [8, 29, 9], which secures web communications;

the 5G-AKA protocol [28], used for authentication in modern mobile networks; the EMV protocol suite [7], which supports billions of payment card transactions worldwide; and the Signal protocol [40], forming the basis of end-to-end encrypted messaging applications used daily by hundreds of millions of users. These case studies demonstrate the ability of symbolic verification tools to scale beyond toy examples and provide meaningful security guarantees for complex systems.

2 A Bird's-Eye View of the State of the Art

This section provides a high-level overview of the main ideas and selected results in the formal verification of security protocols. Rather than aiming for completeness, we focus on key concepts and representative approaches that have shaped the field.

Although cryptographic protocols may be required to satisfy many properties, the earliest theoretical results focus on trace properties (i.e., properties over the sequences of messages that may be exchanged during protocol executions), and in particular on secrecy: some data generated by an honest participant, such as session keys, should not be revealed to the attacker. To verify such properties, the attacker is typically modeled as an active entity with complete control over network communications. In particular, it may eavesdrop on and replay messages, impersonate honest participants, and generate fresh values (nonces). It may also construct new messages from its knowledge using its deduction capabilities. The problem of deciding whether a protocol preserves secrecy against such an active adversary is referred to as the *security problem*.

Results for an unbounded number of sessions

Although cryptographic protocols are often described in a concise way, their verification problem is difficult due to several sources of unboundedness in their modeling, such as the number of sessions, the length of messages, or nonce generation. Several encodings of the Post Correspondence Problem have been proposed (see, e.g., [3]) to establish the undecidability of the security problem. In particular, using nonce generation, it has been shown that the security problem is undecidable even when message length is bounded [36].

Despite these negative results, several decidability results have been obtained under restrictive assumptions. One of the earliest results is a PTIME complexity result for ping-pong protocols between two participants [32]. In such protocols, at each step, one agent applies a sequence of operators to the last received message and forwards the result to the other agent. A large body of work has also focused on the class of tagged protocols, where a distinct tag (i.e., a constant) is added to each ciphertext. The purpose of tagging is to prevent confusion between messages corresponding to different protocol steps. While this assumption alone is not sufficient to obtain decidability, it has been used to derive several positive results, e.g., [44], including more recent developments [22, 37, 25].

Results for a bounded number of sessions

A well-known result from 2001 shows that the security problem becomes decidable (NP-complete) when the number of protocol sessions is bounded [45]. Note that, even under the assumption of a bounded number of sessions (and therefore a bounded number of nonces), designing a decision procedure remains non-trivial. Indeed, under a bounded number of sessions, the labelled transition system (LTS) representing protocol executions, while finite in depth, remains infinitely branching: whenever an honest agent is waiting for an input, the attacker may provide any message that it can deduce from its current knowledge.

All the results presented so far rely on the perfect cryptography assumption, where cryptographic primitives (such as pairing and encryption) are modeled without any algebraic properties. In particular, encryption is treated as a black box: the only way to recover a plaintext from a ciphertext is by possessing the corresponding decryption key. However, many cryptographic constructions rely on equational theories that introduce non-trivial algebraic structure. Prominent examples include modular exponentiation, which underlies many key-exchange protocols, and exclusive-or (XOR), the latter being widely used in lightweight cryptography and low-power devices such as RFID systems due to its efficiency.

Reasoning modulo associative and commutative (AC) operators, and in particular XOR, is challenging in symbolic verification, as it introduces a non-trivial equational theory with rich algebraic properties. The study of XOR in symbolic protocol verification dates back to 2003, with two independent decision procedures for the security problem under a fixed number of protocol executions [24, 21]. Both works were published at LICS 2003 and were later recognized with the LICS Test-of-Time Award in 2023, highlighting their impact on the field. Subsequently, other equational theories have been investigated, leading to both decidability results (e.g., [20]) and undecidability results [30, 31].

While theoretically important, such decidability results do not scale well to realistic deployments. In practice, tools based on these procedures are limited by the small number of protocol sessions they can effectively analyze. Moreover, many of these procedures, particularly those involving algebraic operators such as XOR, have, to the best of our knowledge, never been implemented.

Some existing verification tools

Regarding the two main tools already mentioned in the introduction, ProVerif and Tamarin, they date back to 2001 [10] and 2013 [43], respectively (or even 2008 if one considers Scyther, the ancestor of Tamarin [27]). While these are not the only tools for symbolic verification, they are certainly the two main ones for the analysis of protocols with an unbounded number of sessions.

Both tools support some classes of equational theories, in particular those that can be represented by a convergent rewriting system satisfying the finite variant property [23]. However, experience has shown that handling equational theories involving AC operators, such as XOR, is significantly more challenging. The support for this theory was investigated in the context of ProVerif in 2008 [41, 42] through a dedicated extension. However, this approach operates at the level of Horn clauses and is restricted to protocols that can be expressed using $\oplus$-linear Horn clauses. Moreover, it supports secrecy properties only. Support for XOR was integrated into Tamarin in 2018 [34]. In Tamarin, the corresponding extension supports a broader range of security properties [34]. However, a recently identified soundness issue related to an optimization for XOR led to significant termination problems when the optimization was disabled [39, 38]. As a consequence, the procedure was revised and later extended beyond XOR to support user-defined AC operators [39].

3 25 years of ProVerif

This gap between decidability results and practical requirements has led to the development of semi-decision procedures and sound approximations. Rather than guaranteeing termination, a tool such as ProVerif aims at scalability and practical effectiveness.

ProVerif in 2001

ProVerif does not rely on a bounded-session assumption nor on a general decidability result. Instead, it is based on a simple representation of protocols using Horn clauses, together with an efficient procedure for determining whether a fact is derivable from these clauses. More precisely, ProVerif relies on a resolution algorithm with selection to check derivability. The first version of ProVerif was developed by Bruno Blanchet in 2001 and published at the CSFW conference [10].

`https://proverif.inria.fr`

At that time, ProVerif only handled secrecy properties, and the user had to specify the protocol directly in terms of Horn clauses. The tool may fail to terminate or return "cannot be proved", but when it succeeds in proving a property, the result is sound. Despite its incompleteness, ProVerif has demonstrated strong practical effectiveness, and already in its early versions many protocols from the literature were successfully analyzed with very modest computational resources. This work, first published at CSFW 2001, received the Test-of-Time Award twenty years later in recognition of its impact on the community.

Since its first version, ProVerif has undergone numerous extensions and has achieved significant success on a wide range of real-world protocols. It has been successfully applied to the analysis of TLS 1.3 [8, 9] and secure messaging protocols such as Signal [40]. Its scalability and high level of automation have made it one of the most widely used tools in symbolic protocol verification. Over time, several extensions have further improved the tool, notably the introduction of the applied pi-calculus [1] as a convenient specification language for security protocols, as well as many enhancements to the input language (e.g., private channels, tables, phases, ...). We briefly outline below some of the main advances that have been made for this tool.

Attack reconstruction

While the Horn-clause technique used in ProVerif can prove security properties for an unbounded number of sessions, it initially provided no concrete information when proofs failed. In 2006, an efficient attack reconstruction technique was introduced [2]. The core idea is to exploit derivations of the attacker's knowledge to reconstruct explicit attack traces. Since these derivations rely on abstractions that ignore session counts and synchronization constraints, reconstructing valid executions is non-trivial. When successful, it produces an attack trace corresponding to a real protocol execution. The approach was implemented in ProVerif and constitutes a key feature in the evolution of the tool, significantly improving its ability to produce concrete counterexamples. It was subsequently extended to accommodate the successive developments of ProVerif, including authentication properties and cryptographic primitives defined by equational theories.

From secrecy to privacy-type properties

Very soon after its initial version, the algorithm was extended to handle authentication properties expressed as correspondence queries [11]. A few years later, in 2005, support was added for equivalence-based properties such as anonymity and unlinkability [12]. This extension enabled reasoning about privacy properties formulated in terms of indistinguishability.

Intuitively, two protocols P and Q are indistinguishable if an attacker cannot determine whether it is interacting with P or with Q. This notion constitutes a general framework for expressing privacy-related properties, including anonymity, vote privacy, and unlinkability.

The notion of equivalence used in ProVerif is the so-called diff-equivalence. It is defined over bi-processes, i.e., pairs of processes that share the same structure and differ only in the terms they manipulate. The semantics of bi-processes is given in terms of a reduction relation describing how a bi-configuration may evolve. A bi-process can reduce only when both sides can perform the same step in a synchronised way; for instance, a conditional must evaluate identically on both sides. Otherwise, the bi-process blocks. As expected, this notion of equivalence is stronger than standard trace equivalence used in security definitions, but it is sufficient to establish some security properties. Since its introduction in 2005, several works aimed at extending the applicability of diff-equivalence and improving the analysis of privacy properties such as unlinkability [14, 19, 5], although some limitations remain.

Global states and beyond

The ability of ProVerif to handle protocols with global state has significantly evolved over recent years, starting with the introduction of StatVerif in 2011 [4] and later GSVerif in 2018 [15]. While ProVerif is highly effective for the automatic analysis of security protocols, it has traditionally struggled with constructs involving global state, such as counters, tables, or memory cells, due to its internal abstractions. StatVerif was introduced to address this limitation, but it is restricted to a finite number of cells and may quickly suffer from state explosion. These limitations are overcome by GSVerif. The approach relies on transforming security queries into an equivalent form that can be handled by ProVerif through automatically generated formulas. This transformation was proved sound and enables global-state reasoning to be delegated to automatically derived lemmas. Experimental evaluations show that this front-end, when combined with ProVerif, outperforms existing tools in terms of both efficiency and protocol coverage.

Building on these ideas, a major extension of ProVerif was developed in 2022 to improve the precision of the analysis and reduce inconclusive results [13]. ProVerif was enriched with native support for constructs such as lemmas and axioms, allowing for instance the formulas generated by GSVerif to be directly expressed as axioms and exploited earlier in the verification procedure, thereby increasing the number of successful proofs. The features introduced in [13] not only broaden the scope of the tool but also improve its precision by reducing spurious attacks and enhancing termination behaviour in practice. In addition, several core algorithms of ProVerif were redesigned and optimised, including clause generation, resolution, and subsumption techniques. These improvements resulted in significant performance gains on large-scale case studies.

Beyond the perfect cryptographic assumption

From its early versions, ProVerif has supported a variety of standard cryptographic primitives, including symmetric and asymmetric encryption, digital signatures, hash functions, and a restricted form of modular exponentiation allowing the modelling of Diffie–Hellman key exchange. As highlighted in Section 2, extending verification tools to richer primitives and capturing their algebraic properties – or modelling their inherent weaknesses – has long been a central concern of the community [26].

Recently, several works have further extended the scope of symbolic modelling. In 2023, a refined treatment of hash functions was proposed [16], based on the observation that fine-grained properties of hash functions cannot be faithfully captured using standard equational

reasoning alone. This work led to extensions of both Tamarin and ProVerif, enabling the analysis of protocols that rely on non-idealised assumptions about hash functions and uncovering attacks stemming from discrepancies between symbolic models and concrete implementations. Similar ideas were later applied to the modelling of mix-nets in [35], leading to the discovery of attacks on several electronic voting protocols using ProVerif.

In 2026, an extension of ProVerif was developed to handle the XOR equational theory [17]. This extension integrates algebraic reasoning into the symbolic framework while preserving the automation and scalability of the tool. Although currently limited to trace properties, it already enables the analysis of protocols that were previously beyond the reach of ProVerif. Extending this approach to richer equational theories, such as bilinear pairings, as well as to diff-equivalence properties, remains future work.

4 Conclusion

Formal verification of cryptographic protocols has made significant progress over the past two decades, leading to mature tools such as ProVerif and Tamarin that are now capable of handling complex, real-world protocols. However, several fundamental challenges remain.

Beyond the continuous extension of these tools to support richer classes of protocols and security properties, an important direction concerns usability. Making formal verification accessible to non-expert users remains a key issue, in particular by reducing the modeling effort and improving automation in realistic settings. Equally important is the question of interoperability between tools. Rather than forcing users to choose between different frameworks with complementary strengths, it is desirable to develop unified or interoperable ecosystems that allow results and models to be shared and combined.

Initial steps in this direction have been made with the development of the platform SAPIC+ [18], which aims at bridging different verification approaches. However, much remains to be done to fully integrate symbolic and computational techniques within a common framework. Addressing these challenges is essential to make formal verification both more widely usable and more broadly applicable to the analysis of modern cryptographic systems.

References

1 M. Abadi and B. Blanchet. Analyzing Security Protocols with Secrecy Types and Logic Programs. In *29th Annual ACM SIGPLAN - SIGACT Symposium on Principles of Programming Languages (POPL 2002)*, pages 33–44, Portland, Oregon, January 2002. ACM Press.

2 X. Allamigeon and B. Blanchet. Reconstruction of attacks against cryptographic protocols. In *18th IEEE Computer Security Foundations Workshop, (CSFW-18 2005), 20-22 June 2005, Aix-en-Provence, France*, pages 140–154. IEEE Computer Society, 2005. doi:10.1109/CSFW.2005.25.

3 M. Arapinis. *Sécurité des protocoles cryptographiques: décidabilité et résultats de réduction*. PhD thesis, École normale supérieure de Cachan, France, 2008.

4 M. Arapinis, E. Ritter, and M. Dermot Ryan. StatVerif: Verification of stateful processes. In *Proceedings of the 24th IEEE Computer Security Foundations Symposium, CSF 2011, Cernay-la-Ville, France, 27-29 June, 2011*, pages 33–47. IEEE Computer Society, 2011. doi:10.1109/CSF.2011.10.

5 D. Baelde, A. Debant, and S. Delaune. Proving unlinkability using ProVerif through desynchronised bi-processes. In *36th IEEE Computer Security Foundations Symposium, (CSF'23), Dubrovnik, Croatia, July 10-14, 2023*, pages 75–90. IEEE, 2023. doi:10.1109/CSF57540.2023.00022.

6 M. Barbosa, G. Barthe, K. Bhargavan, B. Blanchet, C. Cremers, K. Liao, and B. Parno. Sok: Computer-aided cryptography. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pages 777–795. IEEE, 2021. doi:10.1109/SP40001.2021.00008.

7 D. A. Basin, R. Sasse, and J. Toro-Pozo. The EMV standard: Break, Fix, Verify. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pages 1766–1781. IEEE, 2021. doi:10.1109/SP40001.2021.00037.

8 K. Bhargavan, B. Blanchet, and N. Kobeissi. Verified models and reference implementations for the TLS 1.3 standard candidate. In *38th IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 483–502. IEEE Computer Society, 2017. doi:10.1109/SP.2017.26.

9 K. Bhargavan, V. Cheval, and Ch. A. Wood. A symbolic analysis of privacy for TLS 1.3 with encrypted client hello. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 365–379. ACM, 2022. doi:10.1145/3548606.3559360.

10 B. Blanchet. An efficient cryptographic protocol verifier based on prolog rules. In *14th IEEE Computer Security Foundations Workshop (CSFW-14 2001), 11-13 June 2001, Cape Breton, Nova Scotia, Canada*, pages 82–96. IEEE Computer Society, 2001. doi:10.1109/CSFW.2001.930138.

11 B. Blanchet. From secrecy to authenticity in security protocols. In Manuel V. Hermenegildo and Germán Puebla, editors, *Static Analysis, 9th International Symposium, SAS 2002, Madrid, Spain, September 17-20, 2002, Proceedings*, Lecture Notes in Computer Science, pages 342–359. Springer, 2002. doi:10.1007/3-540-45789-5_25.

12 B. Blanchet, M. Abadi, and C. Fournet. Automated Verification of Selected Equivalences for Security Protocols. In *20th IEEE Symposium on Logic in Computer Science (LICS 2005)*, pages 331–340, Chicago, IL, June 2005. IEEE Computer Society. doi:10.1109/LICS.2005.8.

13 B. Blanchet, V. Cheval, and V. Cortier. ProVerif with lemmas, induction, fast subsumption, and much more. In *43rd IEEE Symposium on Security and Privacy, SP 2022, San Francisco, CA, USA, May 22-26, 2022*, pages 69–86. IEEE, 2022. doi:10.1109/SP46214.2022.9833653.

14 V. Cheval and B. Blanchet. Proving more observational equivalences with proverif. In David A. Basin and John C. Mitchell, editors, *Principles of Security and Trust - Second International Conference, POST 2013, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2013, Rome, Italy, March 16-24, 2013. Proceedings*, Lecture Notes in Computer Science, pages 226–246. Springer, 2013. doi:10.1007/978-3-642-36830-1_12.

15 V. Cheval, V. Cortier, and M. Turuani. A little more conversation, a little less action, a lot more satisfaction: Global states in proverif. In *31st IEEE Computer Security Foundations Symposium, CSF 2018, Oxford, United Kingdom, July 9-12, 2018*, pages 344–358. IEEE Computer Society, 2018. doi:10.1109/CSF.2018.00032.

16 V. Cheval, C. Cremers, A. Dax, L. Hirschi, C. Jacomme, and S. Kremer. Hash gone bad: Automated discovery of protocol attacks that exploit hash function weaknesses. In Joseph A. Calandrino and Carmela Troncoso, editors, *32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023*, pages 5899–5916. USENIX Association, 2023. URL: https://www.usenix.org/conference/usenixsecurity23/presentation/cheval.

17 V. Cheval and S. Delaune. Symbolic protocol verification modulo XOR in ProVerif. In *Proceedings of the 2026 ACM SIGSAC Conference on Computer and Communications Security, CCS 2026, The Hague, The Netherlands, November 15-19, 2026*. ACM, To appear.

18 V. Cheval, C. Jacomme, S. Kremer, and R. Künnemann. SAPIC+: protocol verifiers of the world, unite! In K. R. B. Butler and K. Thomas, editors, *31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022*, pages 3935–3952. USENIX Association, 2022. URL: https://www.usenix.org/conference/usenixsecurity22/presentation/cheval.

19 V. Cheval and I. Rakotonirina. Indistinguishability beyond diff-equivalence in ProVerif. In *36th IEEE Computer Security Foundations Symposium, CSF 2023, Dubrovnik, Croatia, July 10-14, 2023*, pages 184–199. IEEE, 2023. doi:10.1109/CSF57540.2023.00036.

20 Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. Deciding the security of protocols with Diffie-Hellman exponentiation and product in exponents. In *Proc. 23rd Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS'03)*, volume 2914 of *LNCS*, pages 124–135, Mumbai (India), 2003. Springer-Verlag. doi:10.1007/978-3-540-24597-1_11.

21 Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP decision procedure for protocol insecurity with XOR. In *18th IEEE Symposium on Logic in Computer Science (LICS 2003), 22-25 June 2003, Ottawa, Canada, Proceedings*, pages 261–270. IEEE Computer Society, 2003. doi:10.1109/LICS.2003.1210066.

22 R. Chrétien, V. Cortier, and S. Delaune. Decidability of trace equivalence for protocols with nonces. In C. Fournet, M. W. Hicks, and L. Viganò, editors, *IEEE 28th Computer Security Foundations Symposium, CSF 2015, Verona, Italy, 13-17 July, 2015*, pages 170–184. IEEE Computer Society, 2015. doi:10.1109/CSF.2015.19.

23 H. Comon-Lundh and S. Delaune. The finite variant property: How to get rid of some algebraic properties. In Jürgen Giesl, editor, *Proc. 16th International Conference on Rewriting Techniques and Applications (RTA'05)*, volume 3467 of *LNCS*, pages 294–307, Nara, Japan, 2005. Springer. doi:10.1007/978-3-540-32033-3_22.

24 H. Comon-Lundh and V. Shmatikov. Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. In *18th IEEE Symposium on Logic in Computer Science (LICS 2003), 22-25 June 2003, Ottawa, Canada, Proceedings*, page 271. IEEE Computer Society, 2003. doi:10.1109/LICS.2003.1210067.

25 V. Cortier, A. Dallon, and S. Delaune. A small bound on the number of sessions for security protocols. In *35th IEEE Computer Security Foundations Symposium, CSF 2022, Haifa, Israel, August 7-10, 2022*, pages 33–48. IEEE, 2022. doi:10.1109/CSF54842.2022.9919670.

26 V. Cortier, S. Delaune, and P. Lafourcade. A survey of algebraic properties used in cryptographic protocols. *J. Comput. Secur.*, 14(1):1–43, 2006. doi:10.3233/JCS-2006-14101.

27 C. Cremers. The Scyther tool: Verification, falsification, and analysis of security protocols. In A. Gupta and S. Malik, editors, *Computer Aided Verification, 20th International Conference, CAV 2008, Princeton, NJ, USA, July 7-14, 2008, Proceedings*, Lecture Notes in Computer Science, pages 414–418. Springer, 2008. doi:10.1007/978-3-540-70545-1_38.

28 C. Cremers and M. Dehnel-Wild. Component-based formal analysis of 5G-AKA: Channel assumptions and session confusion. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society, 2019. URL: https://www.ndss-symposium.org/ndss-paper/component-based-formal-analysis-of-5g-aka-channel-assumptions-and-session-confusion/.

29 C. Cremers, M. Horvat, J. Hoyland, S. Scott, and T. van der Merwe. A comprehensive symbolic analysis of TLS 1.3. In B. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1773–1788. ACM, 2017. doi:10.1145/3133956.3134063.

30 S. Delaune. An undecidability result for AGh. *Theoretical Computer Science*, 368(1-2):161–167, 2006. doi:10.1016/j.tcs.2006.08.018.

31 S. Delaune, P. Lafourcade, D. Lugiez, and R. Treinen. Symbolic protocol analysis in presence of a homomorphism operator and *Exclusive Or*. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, Lecture Notes in Computer Science, pages 132–143. Springer, 2006. doi:10.1007/11787006_12.

32 D. Dolev, S. Even, and R. M. Karp. On the security of ping-pong protocols. In *Proc. Advances in Cryptology (CRYPTO'82)*, pages 177–186, Santa Barbara (California, USA), 1982.

33 D. Dolev and A. Chi-Chih Yao. On the security of public key protocols (extended abstract). In *22nd Annual Symposium on Foundations of Computer Science, Nashville, Tennessee, USA, 28-30 October 1981*, pages 350–357. IEEE Computer Society, 1981. doi:10.1109/SFCS.1981.32.

34 J. Dreier, L. Hirschi, S. Radomirovic, and R. Sasse. Automated unbounded verification of stateful cryptographic protocols with exclusive OR. In *31st IEEE Computer Security Foundations Symposium, CSF 2018, Oxford, United Kingdom, July 9-12, 2018*, pages 359–373. IEEE Computer Society, 2018. doi:10.1109/CSF.2018.00033.

35 J. Dreier, P. Lafourcade, and D. Mahmoud. Shaken, not stirred - automated discovery of subtle attacks on protocols using mix-nets. In Davide Balzarotti and Wenyuan Xu, editors, *33rd USENIX Security Symposium, USENIX Security 2024, Philadelphia, PA, USA, August 14-16, 2024*. USENIX Association, 2024. URL: https://www.usenix.org/conference/usenixsecurity24/presentation/dreier.

36 N. Durgin, P. Lincoln, J. Mitchell, and A. Scedrov. Undecidability of bounded security protocols. In *Proc. Workshop on Formal Methods and Security Protocols (FMSP'99) - FLOC Workshop*, Trento (Italy), 1999.

37 S. B. Fröschle. Leakiness is decidable for well-founded protocols. In R. Focardi and A. C. Myers, editors, *Principles of Security and Trust - 4th International Conference, POST 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015, Proceedings*, Lecture Notes in Computer Science, pages 176–195. Springer, 2015. doi:10.1007/978-3-662-46666-7_10.

38 Tamarin GitHub. Bug in built-in XOR (issue 770). https://github.com/tamarin-prover/tamarin-prover/issues/770, 2025.

39 É. Klein. *Formal Verification in Practice: Real-World Case Study and Enhanced Support for AC Operators in Tamarin*. PhD thesis, University of Lorraine, Nancy, France, 2025.

40 N. Kobeissi, K. Bhargavan, and B. Blanchet. Automated verification for secure messaging protocols and their implementations: A symbolic and computational approach. In *2017 IEEE European Symposium on Security and Privacy, EuroS&P 2017, Paris, France, April 26-28, 2017*, pages 435–450. IEEE, 2017. doi:10.1109/EuroSP.2017.38.

41 R. Küsters and T. Truderung. Reducing protocol analysis with XOR to the XOR-free case in the Horn theory based approach. In P. Ning, P. F. Syverson, and S. Jha, editors, *Proceedings of the 2008 ACM Conference on Computer and Communications Security, CCS 2008, Alexandria, Virginia, USA, October 27-31, 2008*, pages 129–138. ACM, 2008. doi:10.1145/1455770.1455788.

42 R. Küsters and T. Truderung. Reducing protocol analysis with XOR to the XOR-free case in the Horn theory based approach. *J. Autom. Reason.*, 46(3-4):325–352, 2011. doi:10.1007/S10817-010-9188-8.

43 S. Meier, B. Schmidt, C. Cremers, and D. A. Basin. The Tamarin prover for the symbolic analysis of security protocols. In N. Sharygina and H. Veith, editors, *Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings*, volume 8044 of *Lecture Notes in Computer Science*, pages 696–701. Springer, 2013. doi:10.1007/978-3-642-39799-8_48.

44 R. Ramanujam and S. P. Suresh. Tagging makes secrecy decidable for unbounded nonces as well. In *Proc. 23rd Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS'03)*, volume 2914 of *LNCS*, pages 363–374, Mumbai (India), 2003. Springer-Verlag.

45 M. Rusinowitch and M. Turuani. Protocol insecurity with finite number of sessions is NP-complete. In *Proc. 14th Computer Security Foundations Workshop (CSFW'01)*, pages 174–190, Cape Breton (Canada), 2001. IEEE Comp. Soc. Press.