Tamarin: Verification of Large-Scale, Real World, Cryptographic Protocols

David Basin, Cas Cremers, Jannik Dreier, Ralf Sasse

Tamarin: Verification of Large-Scale, Real World, Cryptographic Protocols

February 16, 2022

David Basin
ETH Zurich

Cas Cremers
CISPA Helmholtz Center for Information Security, Saarland University

Jannik Dreier
Université de Lorraine, CNRS, Inria, LORIA

Ralf Sasse
ETH Zurich

***Abstract*—Tamarin is a mature, state-of-the-art tool for cryptographic protocol verification. We introduce Tamarin and survey some of the larger, tour-de-force results achieved with it. We also show how Tamarin can formalize a wide range of protocols, adversary models, and properties, and scale to substantial, real-world, verification problems.**

Introduction

Cryptographic protocols provide a basis for secure computing in distributed environments. We use these protocols daily, often without much thought. For example, we use TLS every time our browser securely connects to a webserver on the Internet or we download software patches. We use IPSec to setup virtual private networks and SSH for secure remote login. Behind the scenes we may be using Kerberos, OAuth2, or OpenID Connect for single sign-on or access delegation. And when we make payments with our credit card, perhaps stored on our smart phone, we are using the EMV protocol of Europay, Mastercard, and Visa. As these examples suggest, cryptographic protocols are used in critical applications and hence they must operate correctly. Formal Methods and associated tools play an essential role in ensuring that they do so.

Verification tools for cryptographic protocols have been under development since the 1980s. The tools employ algorithmic verification techniques ranging from traditional model-checkers and search procedures to constraint solvers. Many of the tools developed differ from classical model checkers in that they handle undecidable verification problems. They work with protocol models that are infinite state and include formalizations of cryptographic operators using equations. A history of these tools and the central ideas behind them can be found in [1].

In this article we describe Tamarin, which is an open-source analysis tool for cryptographic protocols. Tamarin takes as input a *model* that specifies the protocol and the capabilities of possible adversaries, and the intended *security properties*. Tamarin provides algorithmic support for searching for a counter-example to the security properties, which thereby represents an attack on the protocol. Alternatively, when no counterexample exists, Tamarin constructs a proof.

Tamarin has proven to be a robust and powerful analysis tool. It has been under development for over a decade and has reached a state of

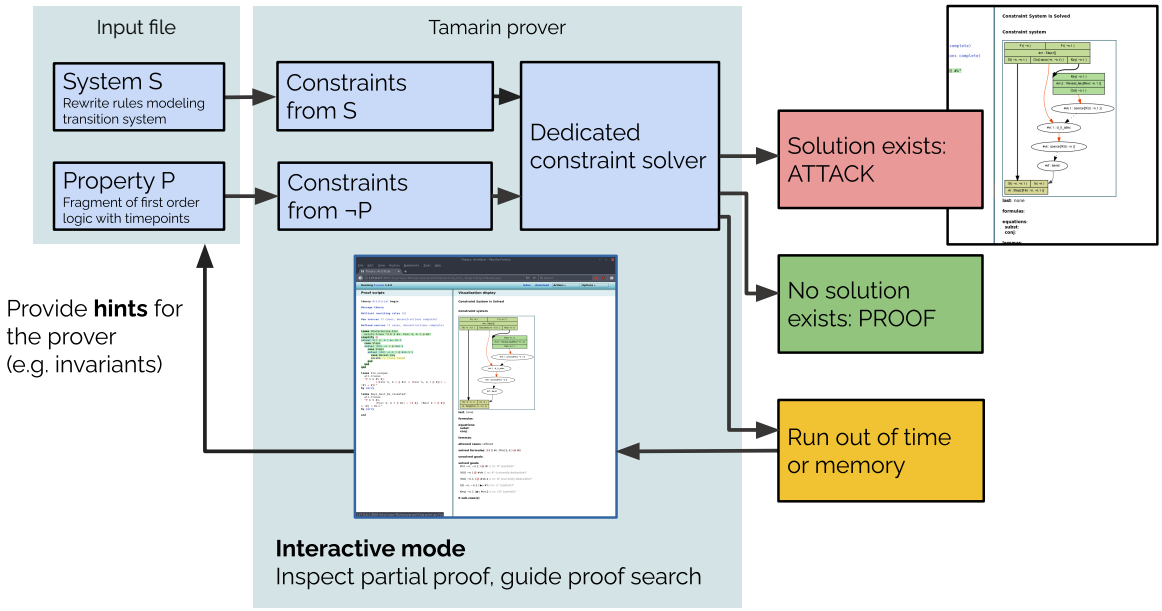

Figure 1. High-level view of interaction with TAMARIN

maturity where it can be applied to model and analyze a wide range of large-scale, state-of-the-art, cryptographic protocols. It is now one of the leading tools in this domain, with a very active user community spanning both academia and industry.

Our focus in this article is on how TAMARIN *scales* in terms of the range of designs and properties that can be modeled and reasoned about, as well as their complexity. We also provide examples, from the growing collection of tour-de-force results, that show TAMARIN's application to some of the most popular and critical cryptographic protocols in use today. These examples also underscore the benefits that TAMARIN provides in finding errors, correcting designs, and more generally supporting the protocol development and standardization process.

Verification using TAMARIN

TAMARIN provides general support for modeling and reasoning about cryptographic protocols. Protocols and adversaries are specified using an expressive language based on multiset rewriting rules. These rules define a labeled transition system whose state consists of a symbolic representation of the adversary's knowledge, the messages on the network, information about freshly generated values, and the protocol participants' local states. The adversary and the protocol interact by updating network messages and generating new messages. TAMARIN also supports the equational specification of various cryptographic operators, such as Diffie-Hellman exponentiation, exclusive-or, and bilinear pairings. Security properties are modeled as trace properties, checked against the traces of the transition system, or in terms of the observational equivalence of two transition systems.

Foundations

A formal treatment of TAMARIN's foundations is given by Schmidt, Meier, Cremers, and Basin [2], [3] and here we just recount some of the key technical ideas. For an equational theory E defining cryptographic operations, a multi-set rewriting system R defining a protocol, and a formula ϕ defining a trace property, TAMARIN can either check the validity or the satisfiability of ϕ for the traces of R modulo E. Formulas are expressed in a fragment of first-order logic with quantification over timepoints. As usual, validity checking is reduced to checking the satisfiability of the negated formula.

To check satisfiability, TAMARIN employs constraint solving to perform an exhaustive, symbolic

search for executions with satisfying traces. The states of the search space are represented as constraint systems. For example, a constraint can express that some multi-set rewriting step occurs in an execution or that one step occurs before another. We can also directly use formulas as constraints to express that some behavior does not occur in an execution. Applications of constraint-reduction rules, such as simplifications or case distinctions, correspond to the incremental construction of a satisfying trace. If no further rules can be applied and no satisfying trace was found, then no satisfying trace exists and one has a proof of the protocol's security.

When manipulating constraints, TAMARIN exploits the finite variant property to reduce reasoning modulo E with respect to R to reasoning modulo AC with respect to the variants of R using folding variant narrowing. Moreover, the most recent version of TAMARIN only requires that user-specified equational theories are convergent and ensure the finite variant property [4]. This enables TAMARIN to work with a very large class of equational theories.

As practical examples, the above features enable TAMARIN to handle: protocols with non-monotonic mutable global state and complex control flow such as loops; complex security properties such as the eCK model for key exchange protocols; and equational theories such as Diffie-Hellman, exclusive-or, bilinear pairings, and convergent user-specified theories with the finite variant property.

The verifiability of security protocols is an undecidable problem. There are different independent reasons for this. For instance, an unbounded number of sessions with an active adversary leads to an undecidable reachability problem, e.g., can protocol execution reach a state where the adversary learns a nonce or a key? Moreover, adversary deduction, which is the problem of determining what the adversary can learn from messages he has seen, is itself undecidable in the presence of rich equational theories. To mitigate this problem, TAMARIN combines numerous methods. These include the normalization of terms and executions, the use of heuristics for the backwards search enriched with forward reasoning, induction, and pre-computation. Furthermore, TAMARIN also provides the user ways to interact with the prover during proof search, which we discuss next.

Usage

TAMARIN provides two ways to construct proofs. It has an efficient, fully automated mode that combines deduction and equational reasoning with heuristics to guide proof search. Figure 1 depicts how TAMARIN is typically used. If TAMARIN's automated proof search terminates, it returns either a proof of correctness (for an unbounded number of role instances and fresh values) or a counterexample, representing an attack that violates the stated property. For many of the protocols in Table 1, their analysis is fully automatic.

Since the correctness of cryptographic protocols is an undecidable problem, TAMARIN may fail to terminate on a given verification problem. Users may then resort to TAMARIN's interactive mode where they can examine the proof states, inspect attack graphs and even internal pre-computations, and explore reasons for non-termination. The interactive mode also allows users to manually guide proofs, when necessary (entirely or just in parts), and to export proofs, which can be subsequently loaded into TAMARIN. This interactive mode extends TAMARIN's applicability to far more complex protocols than is possible in tools that only support fully-automated proof construction.

Scope of Applications

TAMARIN has been applied to a wide range of protocols. Table 1 provides an overview of some of the previous applications explored by the TAMARIN community. These go far beyond traditional cryptographic protocols and come from diverse domains including distance bounding, e-voting, and secure routing. Below we highlight three success stories, describing larger, impactful applications of TAMARIN. These three examples, TLS 1.3, 5G-AKA, and EMV were each developed by some subset of this paper's authors together with other colleagues.[1] We emphasize though that there are many other impactful examples, not covered here, by other users who have independently applied TAMARIN to ambitious, large-scale protocols, including payment systems [5], e-voting systems [6], and distance bounding protocols [7].

[1] Note that in these three examples, we name all the researchers involved. For simplicity of exposition, we will use the "royal we" when describing the work done.

Key Exchange	**Authentication**	**Protocols with loops**
Naxos	WS-Security	TESLA1
Signed Diffie-Hellman	ACME (Let's Encrypt)	TLS 1.3
Station-to-Station	**Industrial**	IEEE 802.11 WPA2 (WiFi)
KEA+	DNP3-SAv5 (Grid)	5G handover
IKEv2	MODBUS	**Non-monotonic global state**
Wireguard	OPC-UA	Keyserver
PQ-Wireguard	**Distance Bounding**	Envelope
Noise protocol suite	Brands and Chaum	Exclusive secrets
Key Exchange (multiple parties)	Meadows et al.	Contract signing
5G-AKA	Hancke and Kuhn	PKCS#11
Group protocols	Swiss-Knife	YubiKey
GDH	Kim and Avoine	YubiHSM
TAK	**Payment**	Anonymous Attestation
(Sig)Joux	EMV (Chip-and-PIN, contactless)	TPM 2.0
STR	**Vehicular**	**Secure routing**
Identity-based KE	V2X revocation	DRKey (SCION)
RYY	**E-voting (Hyperproperties)**	**PKI**
Scott	Alethea	ARPKI (incl. global state)
Chen-Kudla	Selene	**Transparency**
	Bulletin boards	KUD/DECIM (incl. global state)

Table 1. A selection of protocols that were modeled and analyzed using TAMARIN

After our three examples, we return to the question of TAMARIN's scope. We show, in particular, how TAMARIN can scale to analyzing large-scale *families* of models, where the families cover sets of protocol variants, adversaries, and properties.

TLS 1.3

Our first success story concerns the Transport Layer Security (TLS) protocol, which is probably the most used cryptographic protocol, world over. It underlies all secure Internet connections that use HTTPS, where it represents the 'S', and many other applications that use TLS as their transport protocol. In the web setting, TLS is typically used to establish a unilaterally authenticated secure channel between a client, such as a web browser, and a server hosting a website or service. The TLS protocol is an Internet Engineering Task Force (IETF) standard initially based on the Secure Sockets Layer (SSL) protocol. It has evolved considerably since its first release as TLS 1.0 in 1999, leading to TLS 1.3 defined in RFC 8446 in 2018.

The core TLS protocol is a key exchange protocol that supports numerous modes and options. For example, TLS contains a negotiation mechanism to agree on ciphersuites and options such as mutual or unilateral authentication. Moreover, each such option has many alternatives. The key exchange protocol produces symmetric keys for the transport layer protocol, which uses a symmetric cipher to encrypt and authenticate message payloads.

In addition to this core functionality, TLS also supports starting connections based on shared symmetric keys, resumption and rekeying mechanisms, out-of-band authentication, and even mechanisms to upgrade unilaterally authenticated connections to mutually authenticated ones. Furthermore, new versions of TLS must be backwards compatible with previous versions, while ensuring that parties converge to the most secure option that they both support, even in the presence of a network attacker attempting so-called downgrade attacks.

Tamarin Analysis

TLS versions prior to 1.3 had been developed by engineers with little academic involvement. These older TLS versions were also plagued by numerous security vulnerabilities. When the development of TLS 1.3 started, the IETF reached out to several academic teams to help in its development and to ensure they would achieve the most secure TLS protocol yet.

As part of this wider effort during TLS 1.3's development, we built several TAMARIN models of TLS 1.3, developed by Cremers, Horvat, Hoyland,

Scott, and van der Merwe [8], [9]. This was a challenging process. Many aspects of the standard were initially underspecified and were rapidly changing. Moreover, the protocol's complexity was at the limits of what TAMARIN could handle at the time.

During the standard's development, which involved around 30 draft revisions, we incrementally built models of the standard, as it evolved. The effort involved was several person-months, the majority of which were dedicated to understanding the details of the TLS 1.3 standard under development.

During our analysis of the transition between the 10th and 11th draft of the TLS 1.3 standard, TAMARIN found an attack on the proposed implementation of the "delayed authentication" mechanism to upgrade unilateral connections. The attack applies to clients and servers that use client certificates, and combines three modes: the initial key exchange, the resumption mechanism, and the delayed authentication mode. The attack allows malicious server owners (e.g., a web forum) to impersonate its clients towards other servers (e.g., the client's bank), violating the main goal of the delayed authentication mechanism [8].

Impact and Lessons Learned

The TAMARIN analysis directly helped prevent a broken mechanism for delayed authentication. It also helped to clarify the exact guarantees for mutual agreement on the status of connections and it helped those involved in its standardization to gain confidence in the security of the final standard.

When TAMARIN found the attack described above, the individual modes had already been carefully scrutinized by designers and cryptographers. The attack was missed because it depends on subtle interactions between the modes. Notably, the attack involves at least 18 network messages, uses three modes, and involves the attacker feeding random values from one connection into the other; such interactions are extremely difficult to find by human inspection.

Interaction with IETF was very constructive and the standard was amended with protocol changes on the basis of our work, thereby avoiding the broken mechanism. We performed an in-depth analysis of the near-final standard, showing that it meets its main security properties [9]. Additionally, our analysis revealed several subtle behaviors and helped clarify the exact guarantees that the standard provides, which were then documented in the final standard.

5G-AKA

Our second success story revolves around the 5G-AKA protocol. 5G is the latest generation of mobile communication technology, designed for higher data transmission, lower latency, and improved security. The 5G standard runs over thousands of pages of documentation. The most critical component for its security is 5G-AKA, the 5G key agreement protocol that is used by the mobile *user device* (namely its SIM card) and the customer's *home network* (the service provider one has a contract with) to agree on a shared key. All other keys are derived from this shared key. Hence the protocol's correctness is critical for user's data security, the authenticity of messages and calls they receive, the connections they start, and for billing based on usage (call time or data).

5G-AKA is complex. Its complexity stems not only from the specification's size, but also the different contexts it can be used in. For example, when roaming, the user device may connect to mobile networks (called *serving networks*) different from the service provider. The protocol then connects three parties, rather than just two, where only two parties, the user device and home network, share initial secrets. Other complexities arise due to technological and backwards compatibility constraints. For example, as older SIM cards lack the ability to create randomness, the protocol uses a counter to prevent replay attacks rather than fresh randomness. However, to derive shared keys, both parties' counters must be equal and this requires a resynchronization sub-protocol that is used whenever messages are lost (e.g., in mobile scenarios when one travels through tunnels).

Some of the authors of this article had the opportunity of working with a company that was part of the industrial standardization body 3GPP, responsible for standardizing 5G-AKA. This collaboration gave us access to both the 5G specification and 5G specialists, and our focus was on 3GPP's TS 33.501 document. We built initial models for versions leading up to and including v0.7.1 with promising preliminary results. Unfor-

tunately flaws were introduced in the following version, which we discovered using TAMARIN, and these were subsequently fixed prior to the final version, due to our disclosure. The resulting model with successful verification of properties (except privacy) was then for the protocol from v15.1.0 of Release 15 of TS 33.501. Additionally, we uncovered privacy problems that could not be fixed in the 5G standard as doing so would require a substantial protocol redesign.

Tamarin Analysis

Our verification of 5G-AKA, carried out by Basin, Dreier, Hirschi, Radomirovic, Sasse, and Stettler [10], started with an in-depth reading of the relevant protocol documents, as well as discussions with those involved in its standardization. From there, we extracted an abstract version of the protocol, which we converted into an executable, analyzable TAMARIN model. This involved handling complications that arose in the resynchronization protocol, the modeling of the sequence numbers for that, and the use of exclusive-or operations, support for which was added to TAMARIN shortly before the 5G-AKA verification started by Dreier, Hirschi, Radomirovic, and Sasse [11].

The majority of the effort spent was the several person-months needed to understand the specification; the time needed to subsequently formalize the resulting model was relatively short. Some additional person-months were needed for the verification, in particular writing proof strategies to help automate it. Along the way, we found flaws which we reported to the 3GPP; with one exception, these were subsequently fixed in the standard. The final verification result is with respect to the corrected version.

The flaw in 5G that was not repairable concerned privacy, as previously mentioned. The privacy of the user's identity is violated for the 5G-AKA protocol by a fairly simple replay attack that exploits the resynchronization protocol. A further iteration (6G?) should eliminate counter-based mechanisms to solve this problem. Nevertheless, 5G-AKA is still an improvement over 4G, as in 5G the adversary must be active and send messages to check if a specific user is nearby, whereas in 4G a passive adversary can simply listen to radio traffic and learn all the identifiers of users who are near its attack device (using so-called IMSI catchers).

Follow-up work by Cremers and Dehnel-Wild [12] adapted the models to incorporate a more fine-grained view of the internal parties. This analysis revealed several unstated assumptions in the standard. If those assumptions are not upheld, flaws like incorrect attribution of customers for billing purposes are again possible.

Impact and Lessons Learned

The practical impact of our TAMARIN analysis [10] is that multiple mistakes in 5G-AKA were discovered and corrected. As a result, the protocol now given in the standard provides appropriate authentication and secrecy properties, which was not the case before. The most critical vulnerability found by TAMARIN, which was also fixed, was a protocol error that allowed the attacker to induce confusion between users for the home network, i.e., the data or time that are used and should be billed to customer A would be incorrectly billed to another customer B. This disclosure led the authors and the publication [10] to be admitted to the "GSMA Mobile Security Research Hall of Fame" as CVD-2018 CVD#0012. The disclosure process to this industry consortium was unfortunately less straightforward than for TLS 1.3, where the IETF explicitly solicited academic input. Despite quickly finding the problem after the update from v0.7.1 and providing a fix that was ultimately used, it took months to get it fixed.

5G-AKA demonstrates that complex, large-scale industry protocols are directly within TAMARIN's scope. However, having a direct interface to the standardization body would help to better integrate TAMARIN's usage into the standardization process. As is currently still the case, the authors had to use an external vulnerability disclosure process, and thus the information provided took a long time (over 6 months) until the proposed fix was finally applied, despite multiple intermediate versions being released. Furthermore, co-development of the standards and proofs would accelerate the feedback and improvement process as we were only able to analyze each version after it was made public.

EMV

Our third success story concerns EMV, which is the international standard for (credit) card

payments at points of sale. It is used world over for payments with credit cards such as Mastercard and Visa. Over 80% of all global payments use EMV, up to 98% in many European countries. For payments, each user has an agreement with a bank, receives a payment card, and can use it at merchants. This offers convenience, availability, and hopefully security. EMV supports both a contact version, where the card is inserted into a payment terminal (where a PIN is often needed), and a contactless version, where the card is simply held near the reader. A variation of contactless payments is when a mobile phone simulates a linked physical card.

EMV's complexity comes from the large number of parties supporting the standard, backwards compatibility with the billions of cards that were previously issued (and are difficult to change), and the large number of terminals at merchants where change is also very slow. This means that legacy support must be considered throughout.

As EMV is the worldwide standard in card payments, it is a valuable attack target, and thus verifying its claims to security is desirable. This is especially so given that it is a complex protocol, no formal analysis has been done before, and older versions of the protocol have exhibited different kinds of vulnerabilities. Thus, one may expect to find further issues in EMV requiring improvements.

Tamarin Analysis

Our formalization of EMV, carried out by Basin, Sasse, and Toro-Pozo [13], again started with a careful reading the technical documentation. As we did not have access to experts, like in the 5G case, we instead cross-checked our understanding using real-world transaction logs. In this way, we could create a model that matched both the documentation and actual usage. This modeling process was time-consuming and took over 6 person-months of work. Independently, we developed an app to check that any issues found would actually be exploitable in realistic scenarios.

The models developed [13] included the contact and contactless modes, and many different sub-protocols (due to aforementioned backwards compatibility), as well as the differences between the protocol used by Visa and the one used by Mastercard. For the contact setting, the complexity stems from the 24 different, in parts interworking, protocols and choices like online or offline mode, with or without PIN, different encryptions of the PIN, etc. These include three major categories, SDA, DDA, and CDA, referring to the possible data authentication methods, which result in very different security properties. The protocols also use a wide range of cryptographic machinery including message authentication codes, signatures, exclusive-or, and certificates.

In the contactless case there are 16 different versions, split between the Visa and Mastercard groups. Tamarin found novel attacks in the contactless setting against Visa's protocol due to the lack of authentication on the parameter stating whether a PIN must be entered for high-value transactions (which it should) or not. This attack enabled us to bypass the PIN on transactions with Visa cards above the threshold that requires a PIN (usually 50 Euros due to COVID, and 25 Euros before).

We went further and developed and modeled fixes for this vulnerability and used Tamarin to prove that the fixes suffice to protect card transactions by enforcing PIN use. In additional follow-up work by Basin, Sasse, and Toro-Pozo [14], we found that Mastercard cards are vulnerable as well, due to a confusion attack, as the datagrams sent by the Visa protocol and Mastercard protocol are interchangeable by the developed man-in-the-middle app. We extended the model to allow a mismatch between the payment network brand and card issuer brand, resulting in another 16 models, split between the Visa and Mastercard protocols. We again proposed fixes; during the disclosure process, Mastercard was able to activate another layer of detection in their payment network that provided an alternative approach to eliminating the attack on Mastercard cards.

Impact and Lessons Learned

This work provides yet another example of how Tamarin can be used to find attacks on substantial, important, real-world protocols. Moreover, we showed that the attacks are practically feasible, where we conducted transactions without using the PIN for high value purchases. (Note that for these attacks, we used our own cards, paying for the purchased goods so as to avoid defrauding any merchant or bank.) Symbolic analysis, and

the careful reading of the standards, thus helped detect flaws that were buried in the standard for years.

Finding flaws in such protocols is itself only part of the solution. The responsible parties must also be convinced of their relevance if they are to be sufficiently motivated to actually fix their protocol. Unfortunately, and to our surprise, even with strong evidence produced by exhibiting the attacks on actual payment cards, and showing that they are practical, not all vendors were willing to take the required actions.

Scaling to Families of Protocols

We present a final example, the Noise framework, highlighting how TAMARIN scales to families of protocols. This framework describes a large set of cryptographic protocols based on combinations of Diffie-Hellman key exchanges. The most well-known instance is the Wireguard protocol, which is a VPN that is included in the Linux kernel. There are in principle an unbounded number of distinct possible handshakes and a subset comprising 53 explicitly listed handshakes are given in the specification of Noise. Additionally, unlike most other protocols, for the Noise protocols one distinguishes the level of security achieved after each message, rather than after the whole protocol, and each message can contain a (more-or-less protected) payload.

In general, when analyzing cryptographic protocols, one considers adversaries who have different compromise capabilities. For Noise, some protocols provide higher protection against different sets of adversary capabilities, and this is analyzed in detail by Girol, Hirschi, Sasse, Jackson, Cremers, and Basin [15]. All in all, a comprehensive analysis of Noise requires a huge number of combinations of security properties, adversary capabilities, protocol instances, and steps for which the statement is checked. The number of resulting proof obligations (even after eliminating scenarios subsumed by others) is still around 410,000 for those 53 explicit handshake patterns.

We used a dynamic analysis approach, which reduced the number of TAMARIN lemma evaluations to around 150,000 by using binary search over the (ordered) properties. This yielded fine-grained (per step) results showing the strongest adversary under which each Noise protocol is still secure. Along the way, our analysis showed that some handshake patterns are clearly better than others, whereas others are incomparable; this gives the protocol implementer the opportunity to choose the protocol version that best matches the desired guarantees for their use-case. This can, e.g., be a privacy for security trade-off or vice versa. Considering the large number of protocols, properties, compromise scenarios, and per message results, this kind of systematic analysis is well beyond the scope of previous analyses. Those previous analyses focused on just one (or a couple) Noise patterns, looked into properties after protocol completion only, and did not provide a systematic protocol hierarchy, ranking Noise protocols by their relative security.

Conclusion

Cryptographic protocol analysis tools have come a long way from simple protocols where Alice authenticates Bob to substantial real-world protocols like those described in this article. The scaling has been in terms of the size, scope, and complexity of the protocols, as well as the complexity of the adversary model, the properties considered, and the comprehensiveness of the analysis. The real-world impact has been considerable: TAMARIN's use has progressed beyond the academic user community, and is now also embraced by numerous companies working on both proprietary protocols and public standards.

This scaling has been enabled by progress on numerous fronts. Algorithmic advances in computing with logical constraints and new algorithms for establishing observational equivalence have increased both the scope and size of protocols as well as the properties that TAMARIN can handle. This progress has been driven by increasingly challenging case studies, providing feedback on TAMARIN's limitations and priorities for improvements. At the same time, the success stories have raised the bar in terms of complexity and impact, further driving progress. Finally, although cryptographic protocol verification tools originated in the formal methods community, continued interaction with the cryptography community has helped to improve the level of detail that can now be captured in the protocol models.

There still remains much work ahead. From the

technical perspective, pushing scalability even further remains a challenge. Possibilities here include improved automation using more intelligent and easily programmable proof strategies, support for an even greater range of cryptographic primitives, and enabling the reuse of proofs. Further work is also needed to increase TAMARIN's accessibility, including improvements to its user interface, better documentation, and education.

REFERENCES

1. D. Basin, C. Cremers, and C. Meadows, *Model Checking Security Protocols*. Springer, 2018, ch. 24, pp. 727–762.
2. B. Schmidt, S. Meier, C. Cremers, and D. A. Basin, "Automated analysis of Diffie-Hellman protocols and advanced security properties," in *25th IEEE Computer Security Foundations Symposium, CSF 2012, Cambridge, MA, USA, June 25-27, 2012*. IEEE Computer Society, 2012, pp. 78–94. [Online]. Available: https://doi.org/10.1109/CSF.2012.25
3. S. Meier, B. Schmidt, C. Cremers, and D. A. Basin, "The TAMARIN prover for the symbolic analysis of security protocols," in *Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings*, ser. Lecture Notes in Computer Science, vol. 8044. Springer, 2013, pp. 696–701. [Online]. Available: https://doi.org/10.1007/978-3-642-39799-8_48
4. J. Dreier, C. Duménil, S. Kremer, and R. Sasse, "Beyond subterm-convergent equational theories in automated verification of stateful protocols," in *Principles of Security and Trust - 6th International Conference, POST 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings*, ser. Lecture Notes in Computer Science, vol. 10204. Springer, 2017, pp. 117–140. [Online]. Available: https://doi.org/10.1007/978-3-662-54455-6_6
5. A.-I. Radu, T. Chothia, C. J. Newton, I. Boureanu, and L. Chen, "Practical EMV relay protection," in *43rd IEEE Symposium on Security and Privacy, SP 2022*. IEEE, 2022.
6. S. Baloglu, S. Bursuc, S. Mauw, and J. Pang, "Provably improving election verifiability in Belenios," in *Electronic Voting - 6th International Joint Conference, E-Vote-ID 2021, Virtual Event, October 5-8, 2021, Proceedings*, ser. Lecture Notes in Computer Science, vol. 12900. Springer, 2021, pp. 1–16. [Online]. Available: https://doi.org/10.1007/978-3-030-86942-7_1
7. S. Mauw, Z. Smith, J. Toro-Pozo, and R. Trujillo-Rasua, "Distance-bounding protocols: Verification without time and location," in *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*. IEEE Computer Society, 2018, pp. 549–566.
8. C. Cremers, M. Horvat, S. Scott, and T. van der Merwe, "Automated analysis and verification of TLS 1.3: 0-RTT, resumption and delayed authentication," in *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016*. IEEE Computer Society, 2016, pp. 470–485.
9. C. Cremers, M. Horvat, J. Hoyland, S. Scott, and T. van der Merwe, "A comprehensive symbolic analysis of TLS 1.3," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*. ACM, 2017, pp. 1773–1788.
10. D. A. Basin, J. Dreier, L. Hirschi, S. Radomirovic, R. Sasse, and V. Stettler, "A formal analysis of 5G authentication," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*. ACM, 2018, pp. 1383–1396. [Online]. Available: https://doi.org/10.1145/3243734.3243846
11. 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*. IEEE Computer Society, 2018, pp. 359–373. [Online]. Available: https://doi.org/10.1109/CSF.2018.00033
12. 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. [Online]. Available: https://www.ndss-symposium.org/ndss-paper/component-based-formal-analysis-of-5g-aka-channel-assumptions-and-session-confusion/
13. 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*. IEEE, 2021, pp. 1766–1781. [Online]. Available: https://doi.org/10.1109/SP40001.2021.00037
14. D. Basin, R. Sasse, and J. Toro-Pozo, "Card brand mixup attack: Bypassing the PIN in non-Visa cards by using them for Visa transactions," in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX

Association, Aug. 2021. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/basin

15. G. Girol, L. Hirschi, R. Sasse, D. Jackson, C. Cremers, and D. Basin, "A spectral analysis of Noise: A comprehensive, automated, formal analysis of Diffie-Hellman protocols," in *29th USENIX Security Symposium (USENIX Security 20)*. Boston, MA: USENIX Association, aug 2020. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/girol

David Basin is a full professor in the Department of Computer Science at ETH Zurich. He received his Ph.D. from Cornell University in 1989, and his Habilitation from the University of Saarbrücken in 1996. He is Editor-in-Chief of Springer-Verlag's book series on Information Security and Cryptography and, from 2015—2020, of ACM Transactions on Privacy and Security. He is also the founding director of ZISC, the Zurich Information Security Center, which he led from 2003–2011. He is a Fellow of the ACM and of the IEEE. Contact him at basin@ethz.ch.

Cas Cremers is faculty at the CISPA Helmholtz Center for Information Security, and Honorary Professor at Saarland University, in Saarbrücken, Germany. He received his Ph.D. from Eindhoven University of Technology in 2006. From 2006 to 2013 he was a postdoctoral researcher, and senior researcher and lecturer, at ETH Zurich in Switzerland. In 2013 he moved to the University of Oxford as an Associate Professor. In 2015 he became Professor of Information security at the University of Oxford. In 2018 he joined the CISPA Helmholtz Center for Information Security in Germany. His research covers both formal analysis and applied cryptography. Contact him at cremers@cispa.de.

Jannik Dreier is an associate professor at Université de Lorraine in Nancy, France, working in the LORIA lab. He received his Ph.D. from the University of Grenoble in 2013, and was a postdoctoral researcher at ETH Zurich from 2013 to 2015. He is a co-chair of the French working group on formal methods for security since 2020. Contact him at jannik.dreier@loria.fr.

Ralf Sasse is a senior scientist and lecturer at the Department of Computer Science at ETH Zurich. He received his Ph.D. in computer science from the University of Illinois at Urbana-Champaign in 2012. His research focuses on the theory and practice of cryptographic protocol verification, and particularly on developing tools that assist in this task. Recent work applied this to 5G mobile communication and EMV payment card security. Contact him at ralf.sasse@inf.ethz.ch.