WG3 Kick-off meeting

–Report–

WG leaders: Madalina Erascu and Alicia Villanueva

Context

There are many perspectives and approaches to the interaction between proof systems and program verification. Despite significant advances in formal methods, there remains a huge barrier to adopting formal methods in the industry. The key idea of this event is to bring together industry designers and the formal methods research community to share ideas and experiences on how to improve the tools to reduce the barrier to adoption.

The Dresden WG3 meeting was planned to be a two-day meeting. It was held at the TU Dresden, in Dresden (Germany).

We thank the local organizers for their work and effort that contributed to having a successful event.

The goals of the meeting were to

- bring together members of the different communities (including formal methods, systems engineering and security),
- make formal methods more effective and more accessible to all stakeholders,
- transfer knowledge in terms of expertise and scientific tools across different disciplines and between academia and industry, and
- foster collaborations and build synergies among participants to ease the path to more fruitful results for the Action.

The scientific program included talks by young researchers and also by academic and industrial experts that fostered enriching discussions.

Participation and program

The meeting was announced through the official mailing list `epn-wg3-verif@inria.fr`, the Zulip channel (`https://epn.zulipchat.com`), and the Action webpage (`https://europroofnet.github.io/`).

The meeting was attended by 47 researchers in total. Fifteen participants were funded by the action. Most of the participants were members of the action. About half of the attendees were affiliated to European universities, whereas the rest were industry participants.

The program of this event (shown below) started with the presentation of the COST Action in general and the presentation of the WG3 goals and deliverables. Then, the scientific activity began with the tutorial on using ProVerif for security applications followed by talks related to the use of formal methods in security. The discussions after the talks were summarized at the end of the day. The sessions of the second included talks related to industrial applications and potential applications of formal methods in the industrial context. The afternoon included also academic works on the use of formal specifications and formal analysis of security systems.

Day 1 – March 27, 2024	
Registration, Opening & Presentation (9:00-9:15):	*The EuroProofNet COST Action and WG3 "Program Verification"* by Alicia Villanueva
Hands-on tutorial (9:15-11:15)	*ProVerif* by Vincent Cheval (U. of Oxford, UK)
Talks (11:45-12:45)	*Why formal methods remain inaccessible for most cryptographers* by Georgio Nicolas (KU Leuven, Belgium) *Verification of cryptographic protocols* by Ilias Cherkaoui (Walton Institute, Ireland)
Talks (14:30-15:30)	*Platform-level Formal Verification for Public Sector Trustworthy Computing: Considerations and Challenges* by Andreas Berg (gematik GmbH, Germany) *Building a vendor-agnostic attestation service: Confidential Containers Trustee* by Samuel Ortiz (Rivos Inc, France)
Talks & Conclusion of the day (16:15-18:30)	*Towards Modular Trusted Execution Environments* by Carsten Weinhold (Barkhausen Institut, Germany) *An ephemeral virtual TPM device to allow Remote Attestation for Confidential Virtual Machines* by Angelo Ruocco (IBM Research, Switzerland) *Attestation for Mobile Network* by Ghada Arfaoui (Orange, France) *Attesting the Verticals* by Ian Oliver (U. of Jyväskylä, Finland) *Role of Formal Verification in Next-Generation Mobile Networks* by Ayşe Sayin (Istanbul TU, Turkey)

Day 2 – March 28, 2024	
Talks (9:00-11:00)	*Some industry-relevant use cases* by Thomas Fossati (Linaro, Switzerland) *TLS and TEEs* by Arto Niemi (Huawei, Finland) *Protection Environment* by Ionut Mihalcea (Arm, UK) *Attested CSR* by Hannes Tschofenig (University of Applied Sciences Bonn-Rhein-Sieg, Siemens/Germany)

Talks (11:30-12:30)	*Hardening NVIDIA's Confidential Computing: A Formally Verified Implementation of the SPDM Device Attestation Protocol* by Tobias Reiher (AdaCore, Germany) *A rollercoaster ride on the formal analysis of attested TLS* by Muhammad Usama Sardar (TU Dresden, Germany)
Talks (14:30-16:35)	*Enarx attestation validation with Steward* by Richard Zak (Enarx maintainer, USA) *Klave - Trustless Confidential Computing platform* by Rui Almeida (Klave, UK) *Model checking for security* by Lilia Georgieva (Heriot Watt University, UK) *Interactive theorem proving for protocol verification* by Horatiu Cheval (University of Bucharest, Romania)
Talks (17:00-17:50)	*Towards Logical Specification and Checking of Malicious Capabilities* by Andrei Mogage (Alexandru Ioan Cuza University, Romania) *Formal correctness-proofs of refactorings* by Volker Stolz (Høgskulen på Vestlandet, Norway) *Towards formally-verified remote attestation in SSProve* by Jannik Mähn (Barkhausen Institut, Germany)
Wrapping-up (17:50-18:15):	Setting the agenda for the year

Output

One of the outputs of this event has been to allow participants to actively interact among themselves, building collaborative synergies. Thanks to the generosity of the speakers who shared their lines of research, goals and views of the action, the community had the opportunity to discover common interests that were discussed during breaks and after the meeting.

Regarding the scientific output, first of all, on the meeting webpage, the presentations and recordings are available wherever technically possible (see `https://europroofnet.github.io/wg3-dresden24-program/`). Second, some participants shared their ideas through a document produced collaboratively during the meeting. These ideas are included in this report together with other annotations.

The focus of the meeting was set on fostering the synergies between the security community and the use of proof systems and solvers. Some points (non-exhaustive list) discussed during the meeting were:

- How the proposed/used protocols are currently tested/verified;
- Challenges to be addressed currently or in the near future;
- The use of formal methods/proof systems/solvers in industry:

  - Reasons they are not used. Resources that would help in their adoption.
  - – Information on how are they used in the industry. Difficulties and resources that would help in the process.

- Long-term challenges.

A more specific description of some of the talks, research interests and discussions are summarized below.

Research topics

ProVerif

The slides and recording of the tutorial on using ProVerif[1] in the context of protocol verification given by Vincent Cheval are publicly available on the meeting webpage. The tutorial focused on the symbolic approach to protocol verification using ProVerif, proposing a modeling strategy, identifying challenges and comparing to rewriting-based approaches.

Ilias Cherkaoui presented their approach to the verification of cryptographic protocols by using ProVerif. The slides and the recording of the talk are available on the website. It was remarked that depending on the background it can be natural to start using ProVerif (for instance if one comes from algebraic and $\pi - calculus.Theproblemofpost -$ $quantumcryptographyprotocolswouldbethenextstep.$

Challenges in fostering the use of formal methods in industry

The talk by Georgio Nicolas raised the topic of the challenges of using formal methods by cryptographers. The slides and recording is available on the website. Some ideas that arose during the discussion are mentioned below.

Although teaching material for generic proof assistant use is abundant, that is not the case for crypto-specific tools. Maybe better documentation or recipes for the verification of specific problems might help. It is suggested that a possible reason could be that FM people build tools but don't use them, and that's why is little effort made in the documentation.

A repository with the results of crypto verification would be positive to improve the visibility of approaches. In this direction, micro-publishing of the results would be positive.

Another possibility is to have a collection of which provers are used by verification tools in the context of protocols and security, mentioning also the pros and cons of the choice of the prover.

It is to discover whether the Dedukti framework can contribute to making the verification tools more accessible to domain experts.

Challenges for the Formal Verification in the public sector

Andreas Berg introduced this topic. Gematik is the national agency for digital medicine. Digitalization of everything (health history, prescriptions, shared among specialists when the patient is transferred, etc.) poses a number of challenges. For instance:

[1] `http://proverif.inria.fr`.

- Protect personal medical records at assurance level “high” / “very high”. This is crucial especially if records of millions of citizens are aggregated in an HCC provider’s DC. Moreover, it is important to prevent qualified insider attackers from gaining access to any of the medical records.

- Define a suitable provider/solution certification scheme. Some kind of certification of confidentiality computing is needed, maybe the formality of specification level can be used as a quality metric.

- Address sudo channels better: avoid them altogether. A possibility is to limit computer resource sharing to services evaluated to Trust Domain’s assurance level, but this may reduce the scalability/availability of the service.

- Establish a Trust Domain Provider (gematik) as attestation authority

- Automate (Re-)certification as much as possible

There are some projects addressing some of these challenges. The slides of the talk are available on the meeting website.

An ephemeral virtual TPM device to allow Remote Runtime Attestation for Confidential Virtual Machines (Angelo Ruocco, IBM Research, Zurich)

Attestation is an essential service for confidential Virtual Machines (CVM). Third party entities can use attestation to determine the trustworthiness of CVMs and then take actions based on that. For example, secrets should be injected only if the CVM is in a trusted state, and actions to protect sensitive data can be taken if the CVM moves to an untrusted state at runtime.

Current implementations of CVMs provide attestation services that only attest the CVM state at boot time. While this is a first step, maintaining trust in the CVM through runtime attestation is just as important.

We are therefore presenting the design of a secure, ephemeral virtual TPM, that allows to attest the CVM during its entire lifetime. The design is based on real TPM devices, thus allowing the reuse of existing TPM userspace tools as well as making use of runtime attestation by enabling the Integrity Measurement Architecture (IMA).

The ephemeral characteristic allows it to implement a subset of the TPM specification, but extremely simplifies the threat model, reduces TCB and still allows run-time attestation. The vTPM design is protected from both attacks from the host and the guest, and the code is attested as part of the platform-based boot time attestation.

Attesting the Verticals (Ian Oliver, University of Jyvaskyla, Finland)

Attestation, trust and confidentiality do not exist as isolated topics but must be taken as part of a much larger context or system. In this work, we provide some history on how work initially developed with and at Nokia, Nokia Bell Labs and later the University of Jyväskylä addressed the systems aspects of trust and constructed a remote attestation environment to support these ideas in infrastructures such as 5G, future 6G, ORAN and also end-to-end from supply-chain to run-time.

Verticals such as the medical and railways domain extend the basic IoT-Cloud pattern with challenging requirements including resiliency, safety, latency etc. They also provide the necessity for trust in objects such as data, control plane traffic. Some trust failure modes also need to be investigated and mitigated at run-time by keeping a potentially compromised system running rather than reboot/reinstall.

Chains of trust in these systems may be many originating from multiple sources. These chains of trust do interact forming many cross-references and in some respects a 'web of trust' that provides a very powerful mechanism for establishing and maintaining trust. These cross-references are especially important when working with objects such as network slices which encompass interactions between many containers, processes and between trusted (and untrusted) hosts or platforms

An attestation engine has been actively developed over the past years –this was the former Nokia Attestation Engine– known as Jane (`https://gitlab.jyu.fi/ijoliver/jane`).

The next steps in this work will concentrate on the notion of trust, ontologies and formal models of the objects and trust relationships in systems, tooling, metrics (even dealing with quantum concepts of trust), expanding to further use cases and applications, especially in the safety-critical domain and finally to digital forensics and failure mode handling.

Some industry-relevant use cases

Thomas Fossati shared some industry-relevant use cases for formal verification. Each use case is presented with a context, description, goals and specification. The use cases are related to emerging protocols (SUIT architecture) and to running code for the Arm architecture (automotive software, IIoT, medical devices, . . .).

SUIT manifests contain command sequences that are fetched, decoded and executed by the SUIT Manifest processor. One target is to verify the processor or interpreter. The goal is to guarantee that the protocol follows the specified requirements.

In the context of the RATS Verifier, given input evidence and CoRIM description of reference and endorsed values, we aim to compute the Accepted Claims Set (ACS), a combination of reference state and actual state. The goal is to prove that ACS Evaluation is deterministic.

In the context of the Android Virtualization Framework pVM, the goal is to ensure that the channel between the pVM and TA can be trusted.

The slides and recording of the talk are available on the website.

DICE Protection Environment (Ionut Mihalcea, Arm)

The use of Remote Attestation (RA) relies on the ability to measure and represent the entire software stack via measurements, and on the ability to prove to a relying party the veracity of these measurements. Device Identifier Composition Engine (DICE) fulfills the role of both a root of trust for measurement and for reporting. The DICE specification defines a mechanism for the software running on a DICE-enabled platform to take part in a chain of trust, rooted in a unique device secret provisioned by the manufacturer. The entire software stack is represented by a certificate chain, which the software components can use to attest themselves.

While DICE solves the issues it attempts to tackle, it also creates a number of challenges related to the security and usability of the software layers as defined by the specification. DICE Protection Environment (DPE) is an architectural component currently in the process of standardisation, which aims to mitigate some of these challenges. DPE helps secure the secrets required by the software running on the platform, and allows it to handle DICE-related state more easily.

From a formal verification perspective, it would be beneficial to have a verified implementation of some profile of the two specifications (DICE and DPE). DICE* already exists as an implementation of the first two layers of DICE, in Low*. No equivalent implementation exists for DPE, and given the current standardisation work, a formal model could prove particularly useful.

When considering the value of these formally verified implementations, it is important to take into account the usability and maintainability of the code produced. Since this is critical for deployment in the field, is current tooling good enough to address these concerns?

Usable formal methods

The slides and recording of the "CSR Attestation" talk are available on the meeting website. During the talk, Hannes Tschofenig shared also the complexity of the design of protocols and the convenience of using formal analysis during the initial development phases. Then the problem of CSR attestation and open issues and new examples were presented.

Hardening NVIDIA's Confidential Computing: A Formally Verified Implementation of the SPDM Device Attestation Protocol (Tobias Reiher, AdaCore, Germany)

Numerous high-profile security vulnerabilities have highlighted flaws in communication protocols. Despite differences, they share common threats: potential exploitation of critical infrastructure, supply chain risks affecting millions of devices and difficulty in prevention and mitigation. Root causes vary, from incomplete protocol specifications to logic errors during manual software translation and reliance on unsafe programming languages. Standardized protocols and widespread, quality-assured implementations mitigate risks, but vulnerabilities persist, especially in custom implementations lacking thorough review, formal basis and assistance from formal verification tools.

RecordFlux is a toolset for the creation of verifiable communication protocol implementations. RecordFlux provides a high-level domain-specific language for specifying binary data formats and communication protocols which is precise and expressive enough to generate complex, formally-provable source code automatically. Through the toolset, users can formally verify specifications, generate provable SPARK code, and validate specifications using communication traces and existing implementations.

The formal specification in RecordFlux' language serves as a single source of truth. Due to its abstract nature and its specialized support for binary formats and communication protocols, RecordFlux specifications can be written and understood by domain experts who are not necessarily programmers or verification engineers. The automated correctness proofs performed at the specification level guarantee that the SPARK code that is generated from a valid RecordFlux specification can automatically be proven. It is ensured that the generated code contains no runtime errors like buffer overruns and integer overflows, and key properties – namely the behavior of the program with respect to the specified data format and protocol state machine – are shown to be fulfilled at any time.

In the future, we plan to add the possibility to use higher-level proofs (by using a model checker or a cryptographic protocol verifier) to prove properties that are defined in a RecordFlux specification. This includes generic properties such as safety and liveness, as well as custom protocol-related properties such as state reachability/non-reachability and secrecy. The result is increased assurance that the specified communication protocol meets expectations and is secure.

Symbolic Security Analysis of Attested TLS (Usama, Arto, Hannes, Thomas)

Attested TLS has three different variants, depending on the time of signature of evidence relative to the TLS Handshake protocol:

- Pre-handshake Attestation
- Intra-handshake Attestation
- Post-handshake Attestation

A formal analysis of the most popular pre-handshake attestation, namely Intel's RA-TLS protocol was presented. ProVerif was used for the formal analysis. Challenges faced in this work include:

- Incomplete and outdated specs for RA-TLS
- Very few comments in Inria's TLS formal model
- Incomplete validation of draft 20 artifacts
- A simple extension made the artifacts run for 1 month on a high-end server (ice-lake)

Challenges to be tackled in the future include proposing and verifying the fixed version for RA-TLS without changing the TLS stack.

Enarx & Steward Attestation (Richard Zak, Enarx, USA)

Enarx[2] has its own application, Steward[3], which performs remote attestation for Confidential Computing environments via certificate signing requests (CSR). The CSR contains an Extension with an array of data, including the raw attestation report directly from the CPU, the vendor's CRL, AMD: public key, Intel: TCB report. Enarx is able to cache locally:

- AMD & Intel CRL
- AMD: CPU certificate
- Intel: TCB report

Steward only issues the certificate if the validation checks are successful. This includes (but isn't limited to):

- Report validation
- Ensuring the CPU key can't be revoked
- Ensuring the hash of the Enarx binary was expected

[2] `https://github.com/enarx/enarx`
[3] `https://github.com/enarx/steward`

Once Enarx has the Steward-signed workload, Enarx uses it on behalf of the workload for all TLS connections. Thoughts and ideas inspired by the conference: enabling Steward to provide a nonce to ensure freshness, which would allow support for Enarx and other applications to be used without requiring a whitelist of allowed applications (to ensure no modifications).

Not currently using formal methods. It would be beneficial to have this for Enarx but we lack the expertise.

Towards Logical Specification and Checking of Malicious Capabilities (Andrei Mogage, Dorel Lucanu)

The current work focuses on combining the best ideas from both the formal methods of academia and the cybersecurity industry in one "simple" task: Determining what a program (especially malware) is capable of doing. Instead of relying on technologies such as static analysis, debugging, or sandbox solutions, we propose a formal framework consisting of taint analysis and a specific temporal logic, Tainting-Based Logic, which can be easily integrated into a binary instrumentation tool. This results in a practical instrument that is capable of monitoring every individual action of the program, producing a tainted trace that can be used to formally extract program capabilities. **The trace, along with the checked rule (behavioral pattern, capability pattern), can be seen as a proof of the detected capability.** The evaluation, using real-life malware samples, highlights promising results.

Going up the abstraction stack: attestation and getting cryptographers to use it (Lorenzo Martinico, Univeirsity of Edinburgh, Scotland)

Much of the work presented at the workshop presented sound techniques to model the behavior of trusted hardware and attestation protocols and construct safe implementations. While it's certainly important to ensure that trusted hardware is built on solid foundations through this kind of verification, the degree of specificity these proofs result in is not directly useful or necessary for designing cryptographic protocols that rely on attested execution as a component within a wider system. A proof of security for such protocol will be more interested in the specific behavior of an application rather than the potential flaws within the hypervisor or the attestation flow. Those issues affect all protocols designed to run on these architectures, therefore it is desirable to take a more modular approach to security proofs. The Universal Composability framework, widely used for the design of cryptographic protocols, provides an ideal target for this kind of abstraction: protocol designers can rely on ideal subroutines in their proofs as replacements for real protocols without loss of security. This allows distinguishing the formalisation of protocols built on trusted hardware primitives and the hardware itself into separate proofs (and preferably by distinct teams). Several protocols in the literature rely on the black box formalisation of Intel SGX and similar TEE architectures provided by Pass et al, 2017. In our upcoming publication, first presented at PaveTrust 2023, we argue for a more fine-grained approach to have the final "end-user" ideal functionalities capture the capabilities and attacks exposed by the TEE architecture in use for that protocol

COST EUROPEAN COOPERATION IN SCIENCE & TECHNOLOGY
COST is supported by the Horizon 2020 Framework Programme of the European Union

Conclusion

The main goals of the event were achieved successfully. The summary presented in this report is an excerpt of the shared ideas. The slides and recordings available on the website include more details on the topics discussed, open problems and current challenges in the formal verification of security platforms and protocols.

Finally, let us thank again the contributors to the document, specially Muhammad Usama Sardar, which also coordinated the local organization of the meeting. We also thank the Action Management Committee for letting us have this WG3 meeting, and all the speakers and participants.

A List of participants

- Jakub Acs (AWS – Germany)
- Jamil Al Bouhairi (Capgemini / TU Dresden – Germany)
- Rui Almeida (Klave – UK)
- Ghada Arfaoui (Orange – France)
- Andreas Berg (gematik GmbH – Germany)
- Ilias Cherkaoui (Walton Institute – Ireland)
- Horatiu Cheval (University of Bucharest – Romania)
- Vincent Cheval (University of Oxford – UK)
- Thomas Fossati (Linaro – Switzerland)
- Lilia Georgieva (Heriot Watt University – UK)
- Fabian Hauck (University of Stuttgart – Germany)
- Andjela Labovic (UK)
- Dorel Lucanu (Alexandru Ioan Cuza University – Romania)
- Corentin Machu (AdaCore – Germany)
- Norbert Manthey (Amazon Development Center Germany GmbH – Germany)
- Lorenzo Martinico (University of Edinburgh – Scotland)
- Jannik Mähn (Barkhausen Institut – Germany)
- Ionut Mihalcea (Arm Ltd. – UK)
- Andrei Mogage (Alexandru Ioan Cuza University – Romania)
- João Mota (NOVA School of Science and Technology – Portugal)
- Abdulaziz Musaev (TU Dresden – Germany)
- Georgio Nicolas (KU Leuven – Belgium)
- Arto Niemi (Huawei Technologies Oy – Finland)
- Tim Ohlendorf (gematik GmbH – Germany)
- Ian Oliver (University of Jyväskylä – Finland)
- Samuel Ortiz (Rivos Inc – France)

cost
EUROPEAN COOPERATION IN SCIENCE & TECHNOLOGY

COST is supported by the Horizon 2020 Framework Programme of the European Union

- Fatih Ozkaynak (Firat University – Turkey)
- Goran Piskachev (AWS – Germany)
- Violet Ka I Pun (Western Norway University of Applied Sciences – Norway)
- Tobias Reiher (AdaCore – Germany)
- Michael Roitzsch (Barkhausen Institut – Germany)
- Angelo Ruocco (IBM Research – Switzerland)
- Ayşe Sayin (Istanbul Technical University – Turkey)
- Alexander Senier (AdaCore – Germany)
- Volker Stolz (Høgskulen på Vestlandet – Norway)
- Hannes Tschofenig (Siemens – Austria)
- Alicia Villanueva (Universitat Politècnica de València – Spain)
- Sara Zain (TU Dresden – Germany)
- Wenhui Zhang (Bytedance – USA)
- Richard Zak (Enarx maintainer – USA)
- Clara Waldmann (University of Stuttgart – Germany)
- Carsten Weinhold (Barkhausen Institut – Germany)
- Stefan Köpsell (TU Dresden – Germany)

cost EUROPEAN COOPERATION IN SCIENCE & TECHNOLOGY

COST is supported by the Horizon 2020 Framework Programme of the European Union