

# Towards a Data Centric Approach for the Design and Verification of Cryptographic Protocols

Luca Arnaboldi, Roberto Metere

The 26th ACM Conference on Computer and Communications Security, London - United Kingdom  
 Proceedings of ACM CCS 2019 - DOI: 10.1145/3319535.3363262 - Poster session

## Abstract

We propose MetaCP, a Meta Cryptographic Protocol verification tool, as an automated tool simplifying the design of security protocols through a graphical interface. The graphical interface can be seen as a modern editor of a non-relational database whose data are protocols.

The information of protocols is stored in XML, enjoying a fixed format and syntax aiming to contain all required information to specify any kind of protocol. This XML can be seen as an almost semanticless language, where different plugins confer strict semantics modelling the protocol into a variety of back-end verification languages.

In the paper, we showcase the effectiveness of this novel approach by demonstrating how easy MetaCP makes it to design and verify a protocol going from the graphical design to formally verified protocol using a Tamarin prover plugin.

## What is MetaCP?

- **Meta Cryptographic Protocol** verification tool, enables the prototyping of security protocol from the design to the formal verification in minutes, and more!
- **Protocol Specification and Verification (PSV).**
- All information required to describe the protocol.
- Basic language with syntax and very little semantics.
- The plugins interpret the PSV and confer target language semantics.
- Many interpretations through the plugins to export to multiple formats.

**curious observer**

How can I trust the interpretation of the plugins?

Can you import tool models to MetaCP?

Can I visualise attacks on MetaCP?

**stressed researcher**

You need to show how the semantics interpret the structure.

Unlikely, as PSV captures more information than the model.

It is currently not possible, but future extensions may include it!

Why else should I use this tool?

- Gives you a kickstart to formalisation in your favourite language.
- You can export to many languages.
- No syntax errors, no spelling mistakes.
- Need more? Ask us!

**security**

**graphical editor**

## Diffie-Hellman Key Exchange in MetaCP

The following protocol allows two parties to exchange a secret key to enable secure communication.

save as...

```

XML editor - Diffie-Hellman Key Exchange.psv
<entity id="Bob" name="Bob" desc="Bob">
  <variable id="g" type="constant"></variable>
</entity>
<message id="m-1" from="Alice" to="Bob">
  <knowledge entity="Alice">
    <variable id="g" type="constant"></variable>
  </knowledge>
  <knowledge entity="Bob">

```

verify

```

Terminal - Tamarin Prover Output
/* All well-formedness checks were successful! */
end
summary of summaries:
analyzed: protocol.spthy
executable_protocol (exists-trace): verified (6 steps)
metacp@darkstar:/home/metacp/ $

```

## MetaCP Targets Diverse Audiences

**student**

Finally, I can see the whole protocol and understand it better

**engineer**

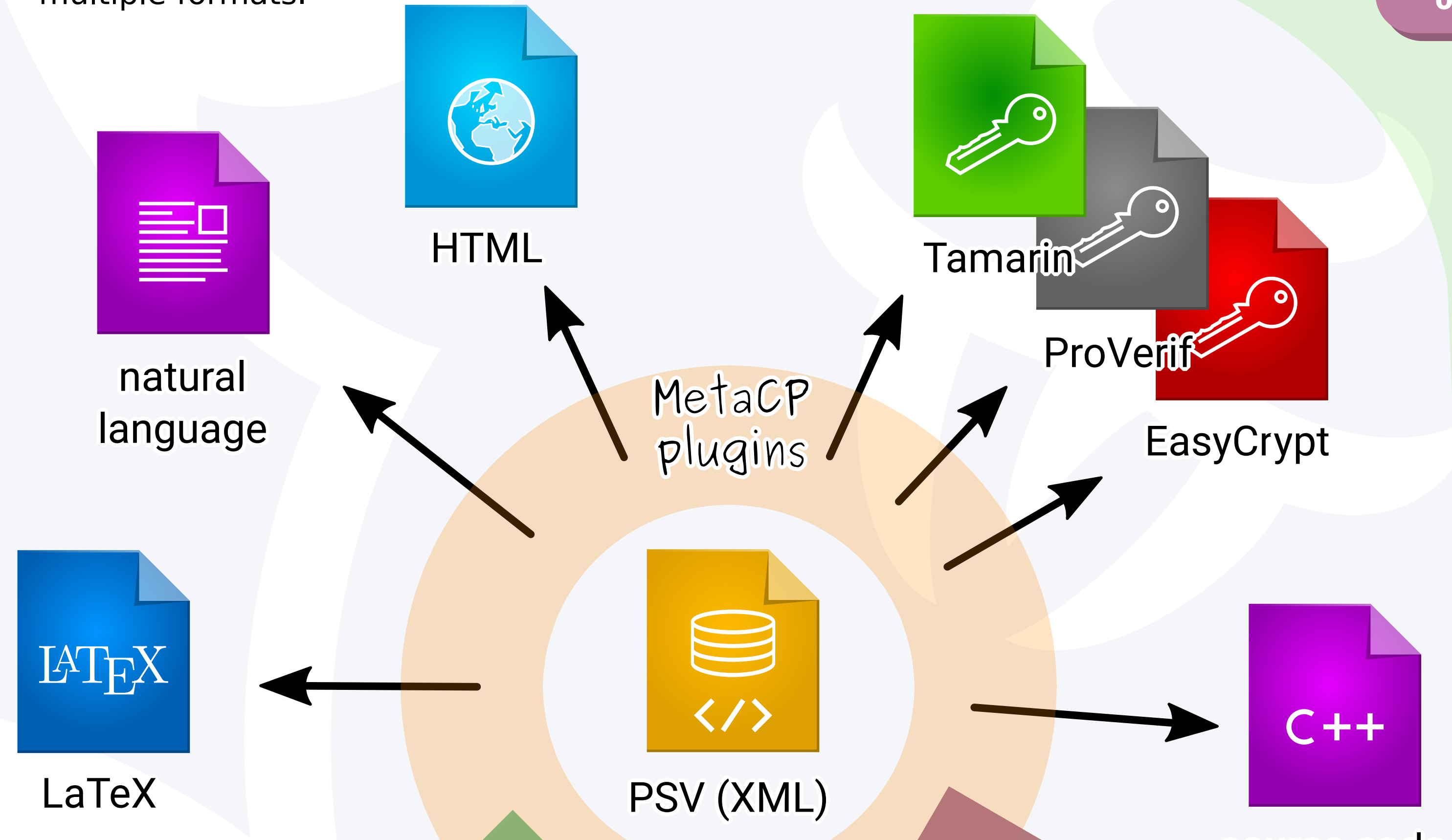
I do not need to understand the cryptography in depth to implement solutions

**security expert**

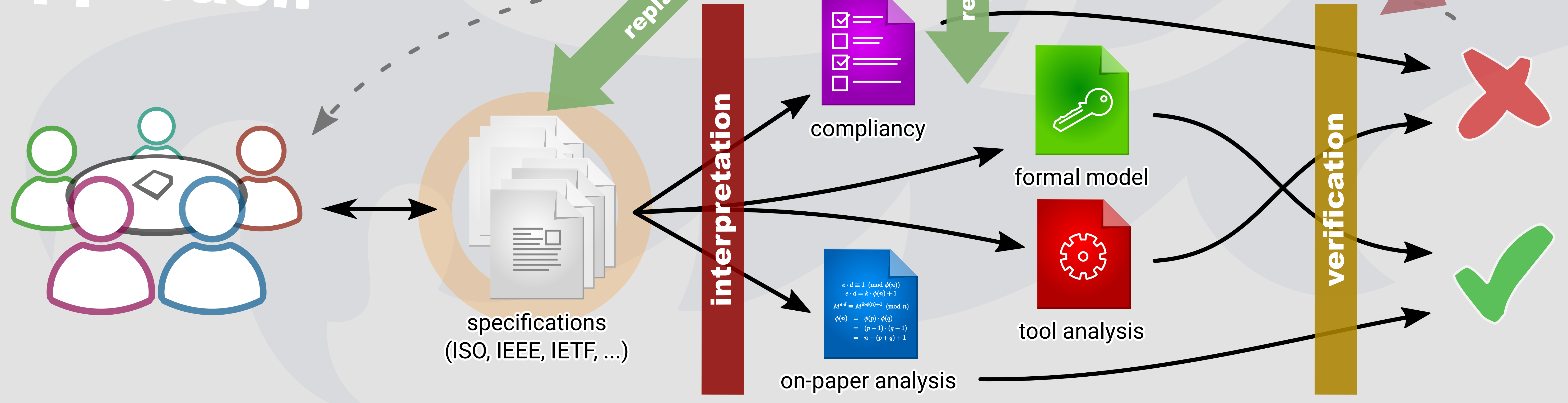
I can prototype solutions and quickly verify subtle mistakes

**formal methodist**

I can reason in multiple models and tools to gain trust in my claims



## current approach



- 1 The MetaCP **Graphical Design Editor (GDE)** aids the design of the protocol.
- 2 The protocol is then saved as a PSV file, which is a structured XML, validated against a DTD specification.
- 3 A plugin for Tamarin code interprets the PSV with Tamarin semantics: the result is coherent code.
- 4 The Tamarin prover is able to verify the code as expected.

