Formally Verified Security

Cătălin Hrițcu

New MPI for Security & Privacy
in Bochum

Formally Verification is Getting Real

Firefox

Mozilla shipping EverCrypt verified crypto library

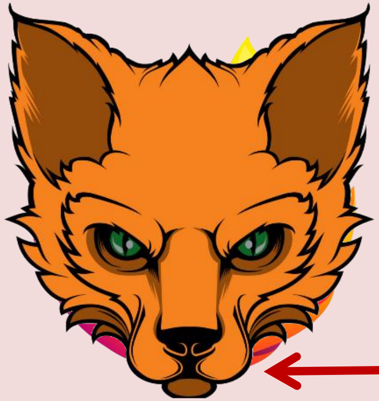
(also used by Microsoft, Linux, ...)

Formal verification milestone:

40.000+ lines of highly-efficient code, proved to be free of vulnerabilities, functionally correct, and side-channel resistant

Formal Verification is Still Limited

Firefox



20.000.000 lines
all unverified

Can we protect?

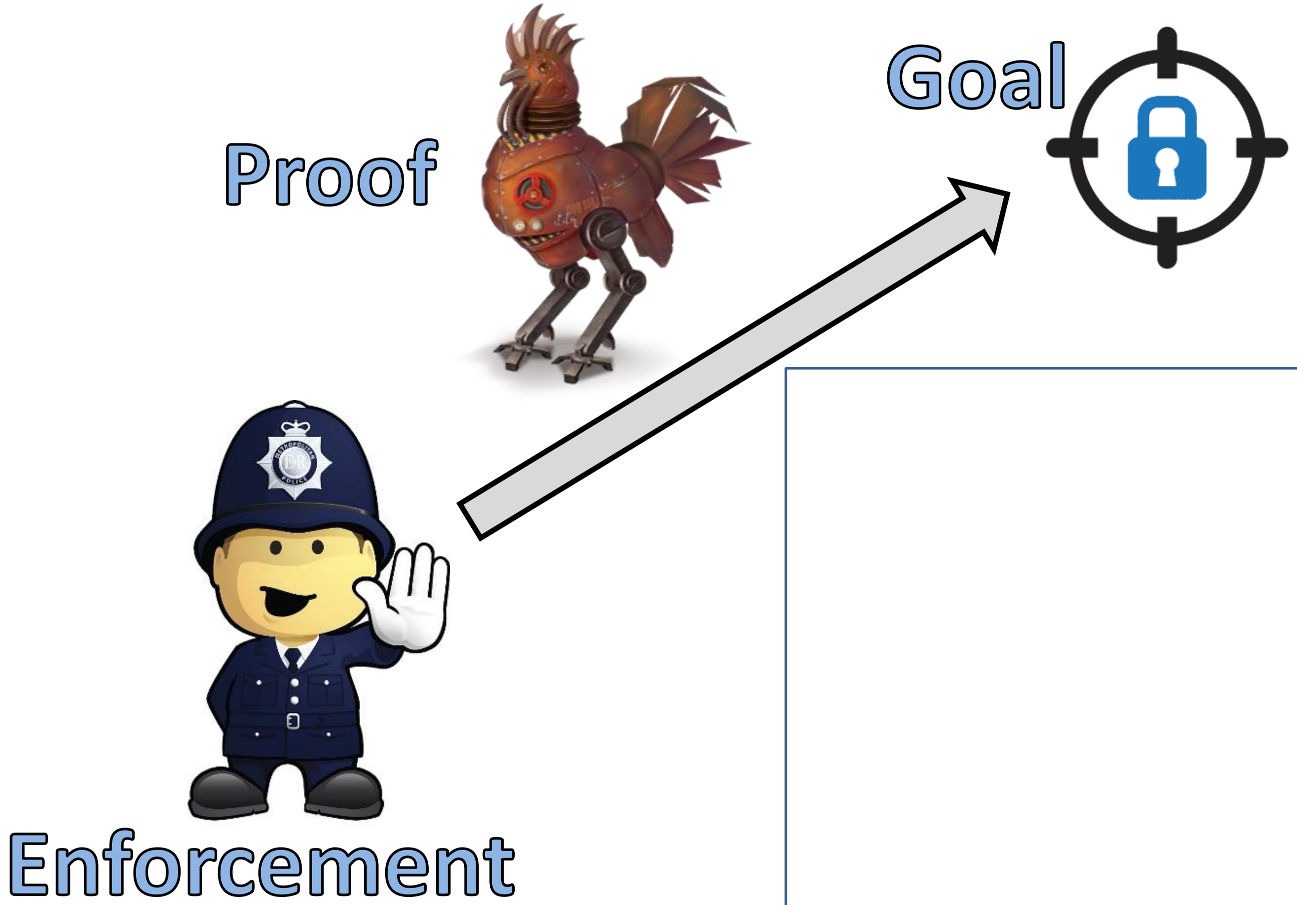
EverCrypt

40.000 lines of F*

insecure interoperability:

if Firefox is compromised it can break security of verified code

Formally Verified Security



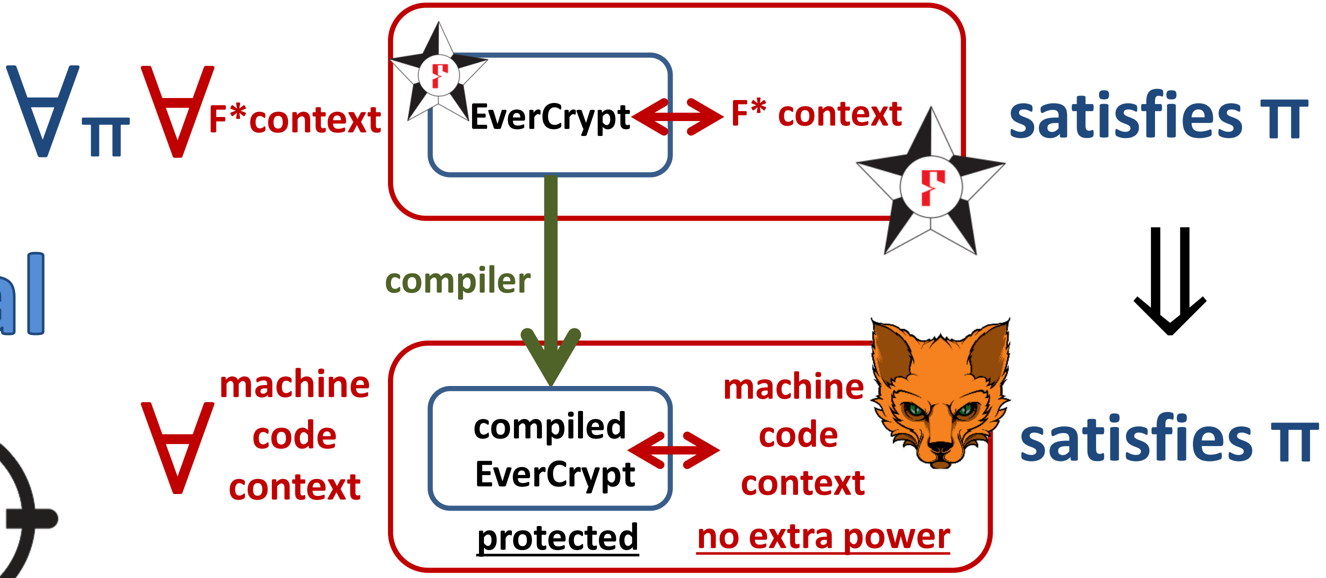
Proof

Goal

Enforcement

1. Security Goal

Formal security definition



$\forall_{\pi}$ $\forall$ F*context

EverCrypt ⟷ F* context

satisfies π

compiler

⇓

$\forall$ machine code context

compiled EverCrypt ⟷ machine code context

satisfies π

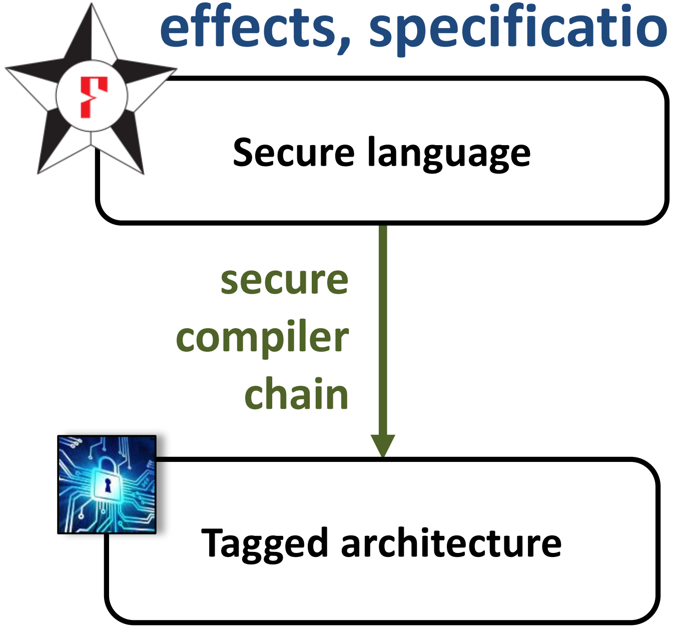
<u>protected</u>

<u>no extra power</u>

2. Security Enforcement

many cool abstractions:
types, modules, functions, effects, specifications, ...

to achieve our goal need to protect these abstractions all the way to machine code



Secure language

secure compiler chain

Tagged architecture

Hardware-accelerated enforcement

3. Security Proof

Formally verifying the security of this compilation chain

- such proofs **very difficult** (wrong conjectures survived for decades)
  and tedious (e.g., 250 pages for toy compiler)
- **more scalable proof techniques**
- **develop proofs as programs**
  - **machine-checked proofs in the Coq proof assistant**

- **simple prototype compiler already verified in Coq**
  - **working on making this realistic**

Formally Verified Security

Cătălin Hrițcu

New MPI for Security & Privacy in Bochum

Denmark
Sweden
Netherlands
Bochum
Germany
Poland
Belgium
Czech Rep.
France
Slovakia
Austria
Switzerland
Italy
Hungary
worldatlas

RUHR REGION OF GERMANY
HAMM
RECKLINGHAUSEN
HERNE
HOTTROP
KIRCHEN
HAUSEN
BOCHUM
DORTMUND
ESSEN
DUISBURG
HAGEN
DÜSSELDORF