Sail Through Your C Code Either Statically or Dynamically with MetAcsl

Virgile Robles, Nikolai Kosmatov,
Virgile Prevosto, Louis Rilling,
Pascale Le Gall

Problem

Function contracts are not suited to express every property:
- Some properties are **hard to express with contracts** alone
- Some properties span across a l**arge number of functions**

Lack of a high-level specification mechanism amenable to automatic verification and testing in FRAMA-C [1].

Example

Confidentiality-sensitive page management:
- each **memory page** has a **confidentiality level**
- each **user** has a confidentiality level
- a process can only **read/write** a page when allowed by the relative confidentiality levels (see Figure 1, 2)
- these constraints are **pervasive** in the program

Solution

A new specification mechanism: the **Meta-Property**
- a set of **target functions**
- a **context** (strong invariant, writing constraint, …)
- a first order **property** on the memory

A verification mechanism:
- translate meta-properties back to ACSL

with the **MetAcsl** [2] plugin for FRAMA-C (see Figure 3, 4).

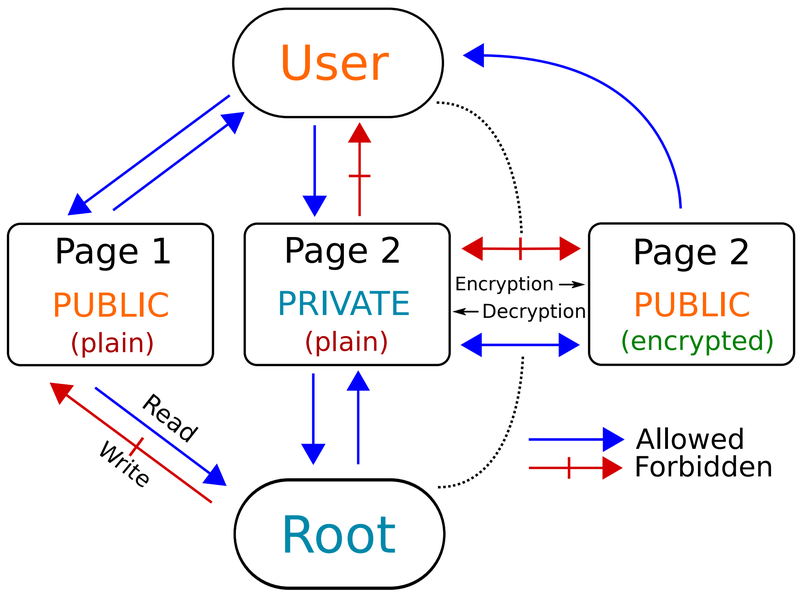

Figure 1: Allowed and forbidden accesses for two agents, two pages and encrypted data

```
struct Page* page_alloc();
void page_free(struct Page* p);
int page_read(
        struct Page* from,
        char* buffer);
int page_write(
        struct Page* to,
        char* buffer);
int page_encrypt(struct Page* p);
int page_decrypt(struct Page* p);
```

Figure 2: Simplified API of the confidentiality-oriented example

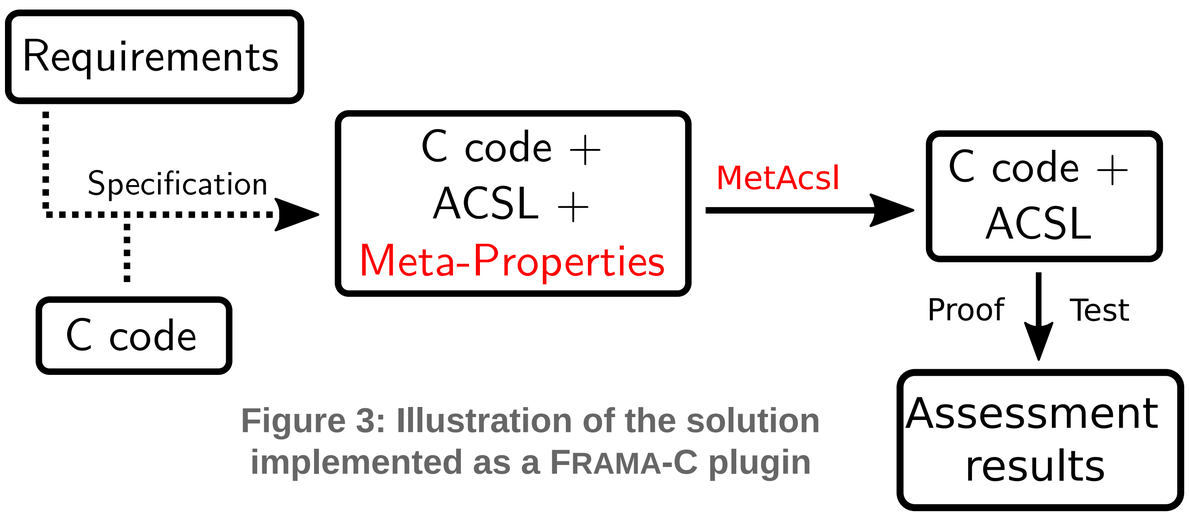

Figure 3: Illustration of the solution implemented as a FRAMA-C plugin

frama C

Software Analyzers

```
/*@ meta \prop,
        \name(write_confidentiality),
        \targets(\ALL),
        \context(\writing),
        \forall Page* p; page_level(p) > process_level
        || \separated(page_data(p)[i], \written);
*/
/*@ meta \prop,
        \name(cons_proc_level),
        \targets(\diff(\ALL, secure_chg_lvl)),
        \context(\writing),
        \separated(&process_level, \written);
*/
```

```
int page_write(struct Page* to, char* buffer) {
        for(int i = 0 ; i < page_size(p) ; ++i) {
                /*@ assert write_confidentiality:
                        \forall Page* p;
                        page_level(p) > process_level
                        || \separated(page_data(p)[i], page_data(to)[i]);
                */
                /*@ assert cons_proc_level:
                        \separated(&process_level, page_data(to)[i]);
                */
                page_data(p)[i] = buffer[i];
        }
}

/* Some other functions */

int page_read(struct Page* from, char* buffer) {
        // ...
        /*@ assert write_confidentiality:
                \forall Page* p;
                page_level(p) > process_level
                || \separated(page_data(p)[i], &process_level);
        */
        //@ assert cons_proc_level: \false;
        process_level = 99999;
}
```

Figure 4: Automatic translation of meta-properties with MetAcsl

Contributions

- A specification mechanism, **meta-properties,** to express high-level properties in Frama-C, and several useful extensions.
- A specification **transformation technique**, enabling the use of existing assessment tools on meta-properties:
  - **Static** deductive verification with the WP plugin
  - **Dynamic** assertion checking with the E-ACSL plugin
- A FRAMA-C plugin, **MetAcsl**, implementing this technique **automatically**, making it easy to re-verify properties after a code or specification update.

References

[1] Kirchner, F., Kosmatov, N., Prevosto, V., Signoles, J., Yakobowski, B.
FRAMA-C - A software analysis perspective.
In: Formal Aspects of Computing (2014)
[2] Robles, V., Kosmatov, N., Prevosto, V., Rilling, L., Le Gall, P.
MetAcsl: Specification and Verification of High-Level Properties.
(tool demo paper) In : TACAS (2019)
[3] Robles, V., Kosmatov, N., Prevosto, V., Rilling, L., Le Gall, P.
MetAcsl : spécification et vérification de propriétés de haut niveau
(long abstract of [2], in French). In: AFADL (2019)

Commissariat à l'énergie atomique et aux énergies alternatives
F-91191 Gif-sur-Yvette Cedex

This work was partially supported by the project VESSEDIA, which has received funding from the EU Horizon 2020 research and innovation programme under grant agreement No 731453. The work of the first author was partially funded by a Ph.D. grant of the French Ministry of Defense.