Formal Analysis of RA-TLS

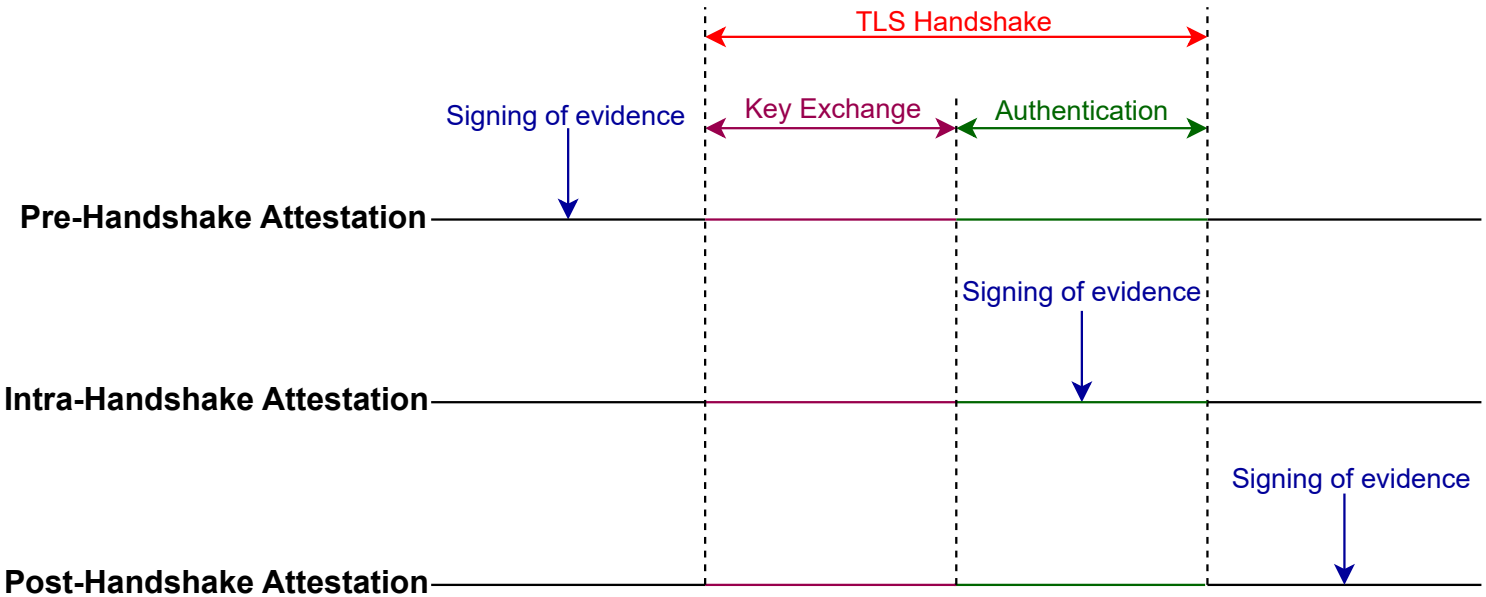
Muhammad Usama Sardar

TU Dresden, Germany

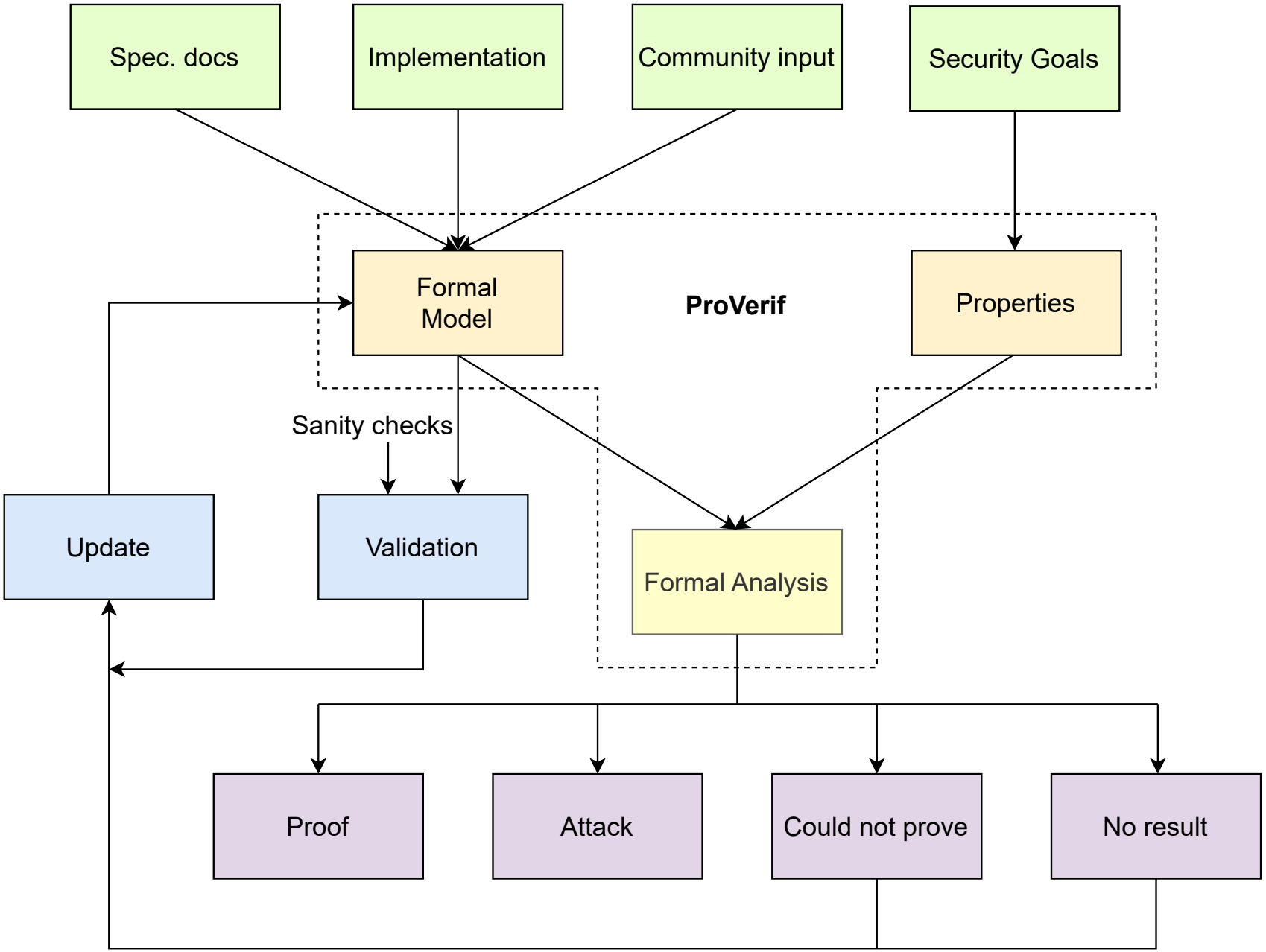
March 18, 2024

Background: Problem in TLS

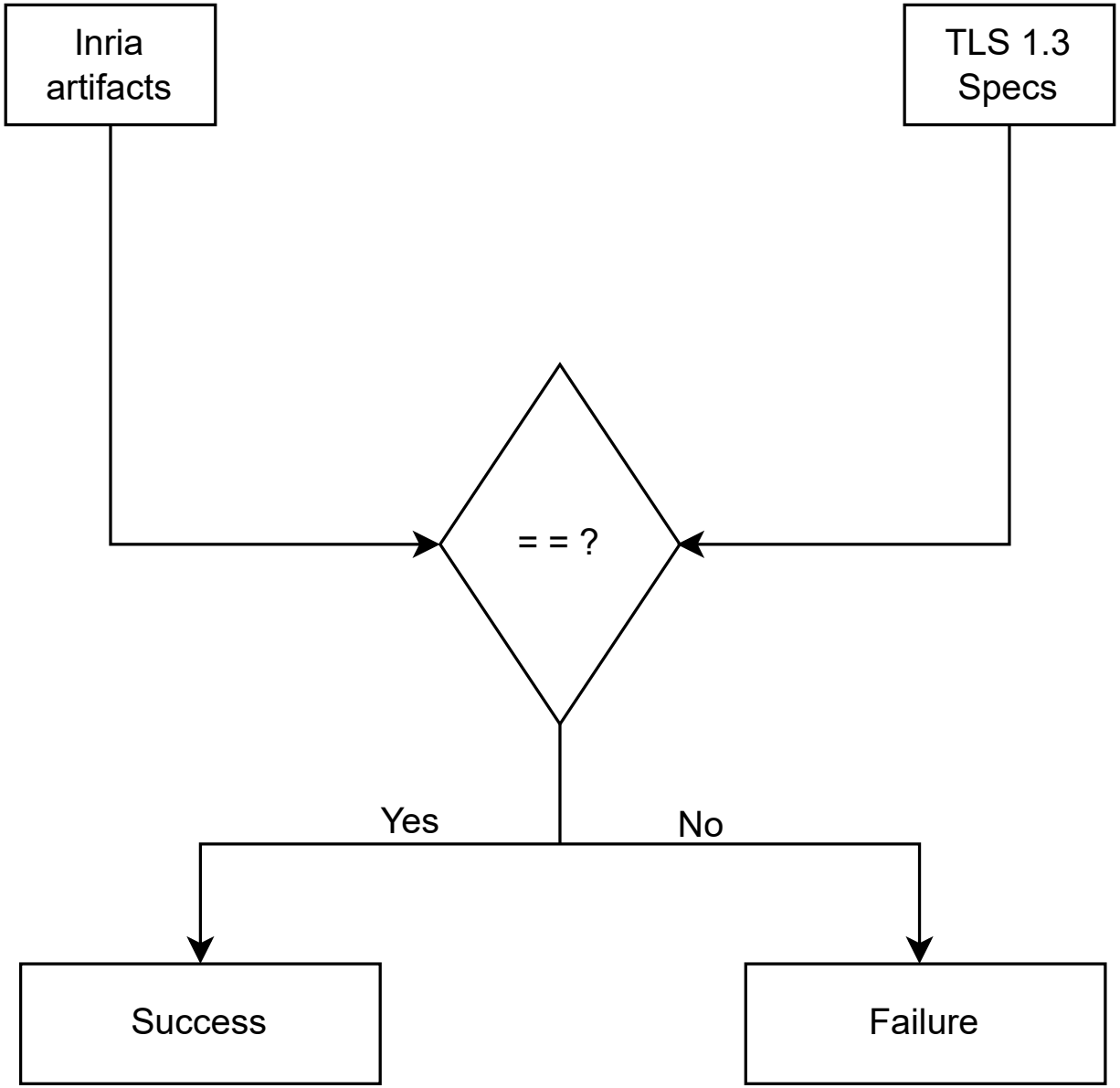
- No validation of security state of endpoint software and platform
  - Need a more comprehensive set of security metrics in some use cases, e.g., CC
- Very complex: at least 15 different exploits
  - Is all complexity (e.g., of key schedule) justified?



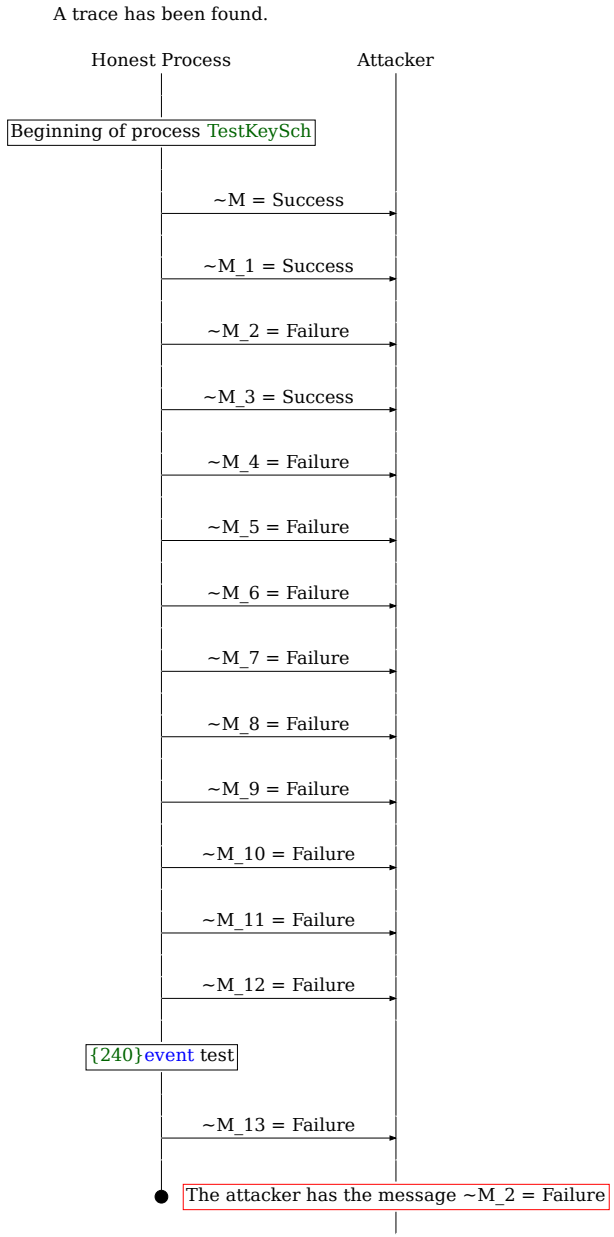
Approach


Spec. docs
Implementation
Community input
Security Goals
Formal Model
ProVerif
Properties
Sanity checks
Update
Validation
Formal Analysis
Proof
Attack
Could not prove
No result

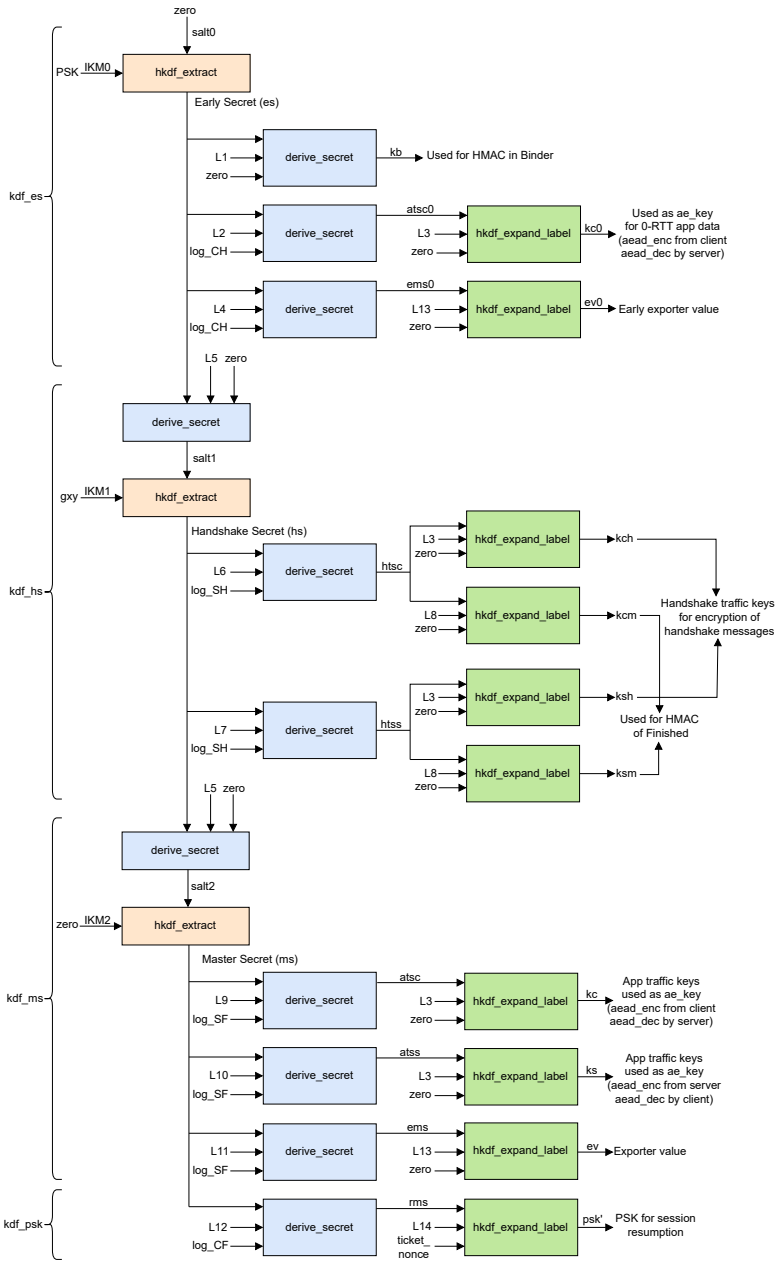
Validation Framework



Validation Result



TLS Key Schedule



Issue 1[1]

Incorrect implementation of salts for Handshake Secret and Master Secret (draft 20 implementation) #7

Open **muhammad-usama-sardar** opened this issue on Dec 4, 2023 · 0 comments

muhammad-usama-sardar commented on Dec 4, 2023

Salt for Handshake Secret

In ProVerif modeling of draft 20, the salt for Handshake Secret derivation is implemented wrongly:

```
let extra = derive_secret(es,tls13_derived,hash(StrongHash,zero)) in
```

Essentially, instead of implementing `Derive-Secret(es, "derived", "")`, the model implements `Derive-Secret(es, "derived", hash(""))`. Since `Derive-Secret` by definition includes hash over Messages, the above formal model results in an additional iteration of hash.

Hence, in accordance with Sec. 7.1 of draft 20, it should be:

```
let extra = derive_secret(es,tls13_derived,zero) in
```

Salt for Master Secret

Same applies to salt for Master Secret:

```
let extra = derive_secret(hs,tls13_derived,hash(StrongHash,zero)) in
```

which should be

```
let extra = derive_secret(hs,tls13_derived,zero) in
```

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
No branches or pull requ

Notifications
U
You're receiving notificati thread.

1 participant

[1] https://github.com/Inria-Prosecco/reftls/issues/7

Issue 2[2]

Incorrect derivation of Master Secret (draft 20 implementation) #6

Open **muhammad-usama-sardar** opened this issue on Dec 4, 2023 · 0 comments

muhammad-usama-sardar commented on Dec 4, 2023

In ProVerif modeling of draft 20, the master secret derivation is implemented wrongly:

```
let ms =   hkdf_extract(hs , zero) in
```

Essentially, the model skips the following step shown in the Key Schedule (cf. diagram showing full key derivation schedule on page 88 in Sec. 7.1 of draft 20):
`Derive-Secret(., "derived", "")`

Hence, it should be:

```
let ms =   hkdf_extract(extra , zero) in
```

[2] https://github.com/Inria-Prosecco/reftls/issues/6

TLS WG[3]

Now about the Inria paper that you have mentioned, I am not much knowledgeable about computational analysis. I understand that it helped them remove the assumption (that DH group elements do not match the corresponding labels) in their proof in CryptoVerif but the corresponding formal analysis in ProVerif in the same paper does not support this view, i.e., all properties remain the same regardless of the additional Derive-Secret.

Moreover, the implementation of key hierarchy in draft 20 in ProVerif by the authors is incorrect [5-6]. For instance, due to a strange reason and beyond our understanding, the draft 20 implementation does not use the Derive-Secret for Master Secret [5]. Do you have any thoughts/opinion on this? The same implementation is being used by other extensions as a baseline, including Lurk [7].

[3] https://mailarchive.ietf.org/arch/msg/tls/ZGmyHwTYh2iPwPrirj_rkSTYhDo/