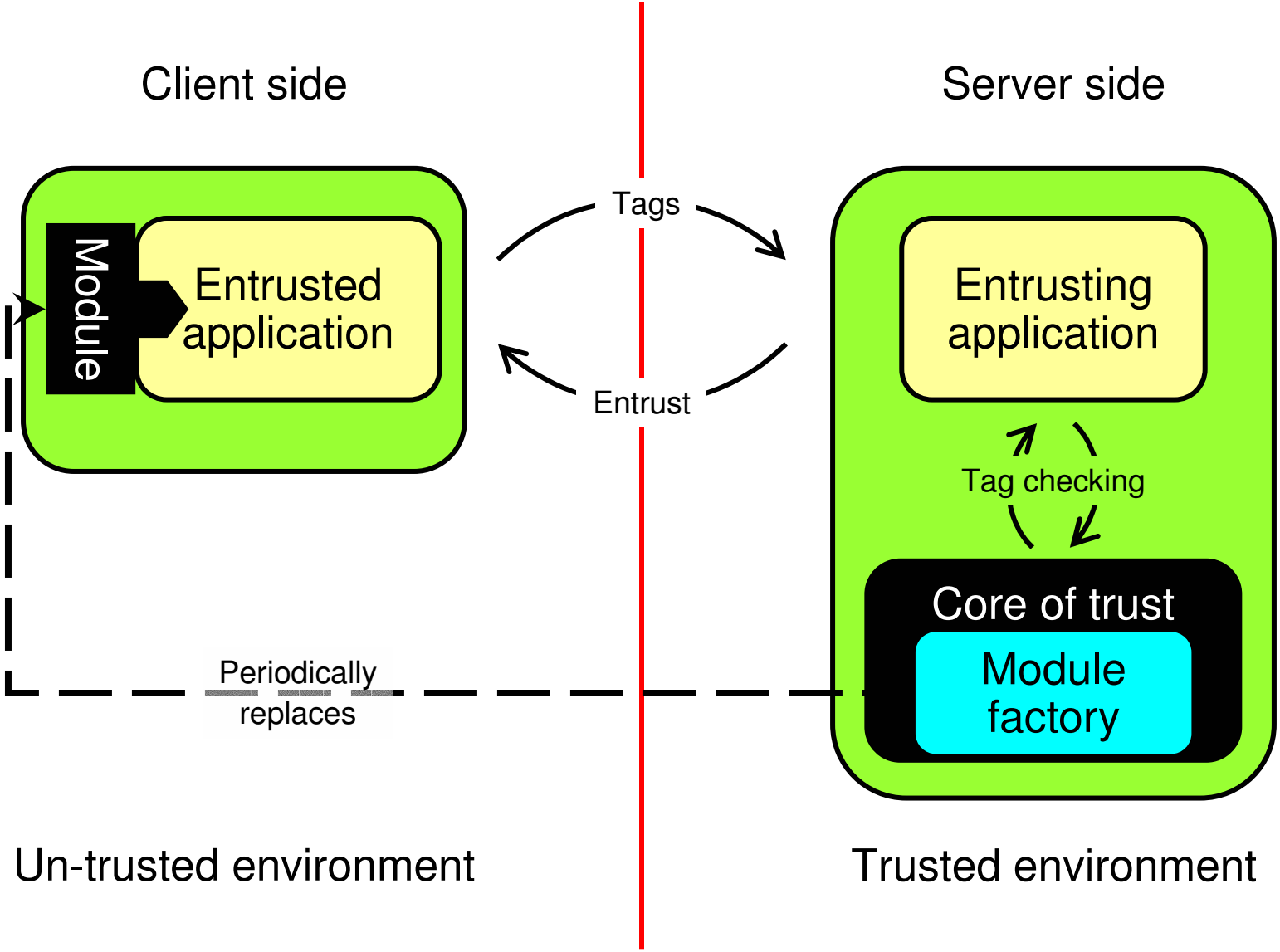
Reverse engineering attacks to remote software entrusting

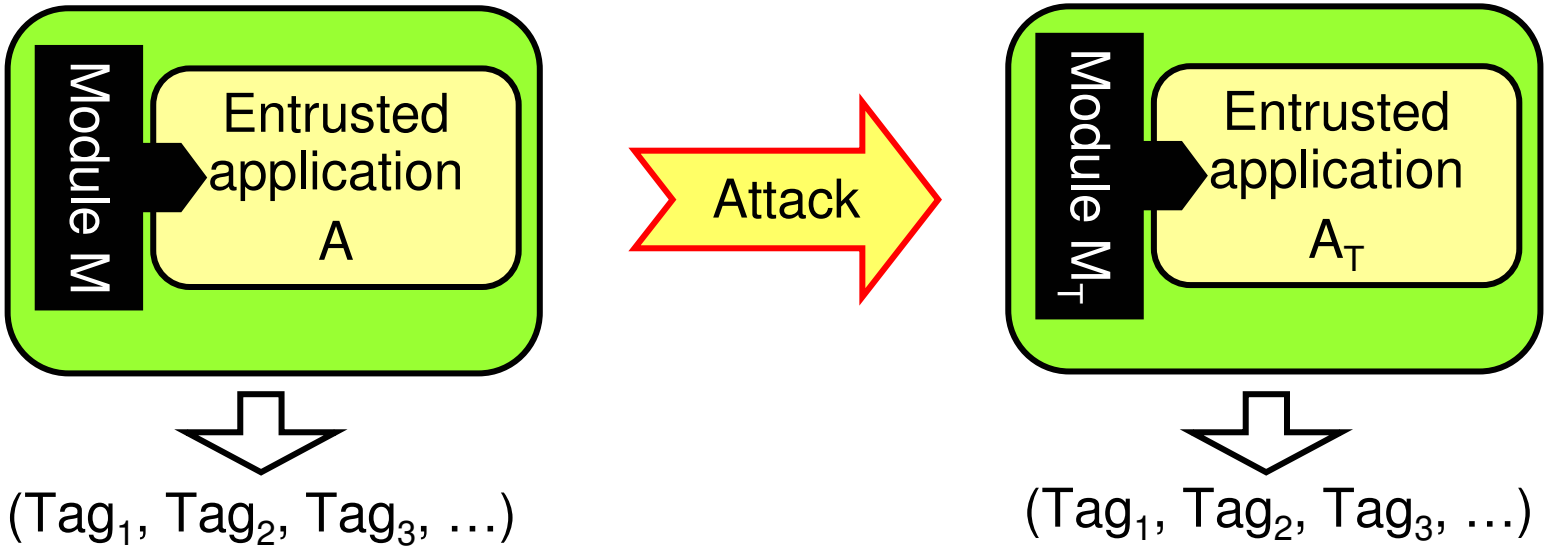
Mariano Ceccato
ceccato@itc.it

Reference scenario



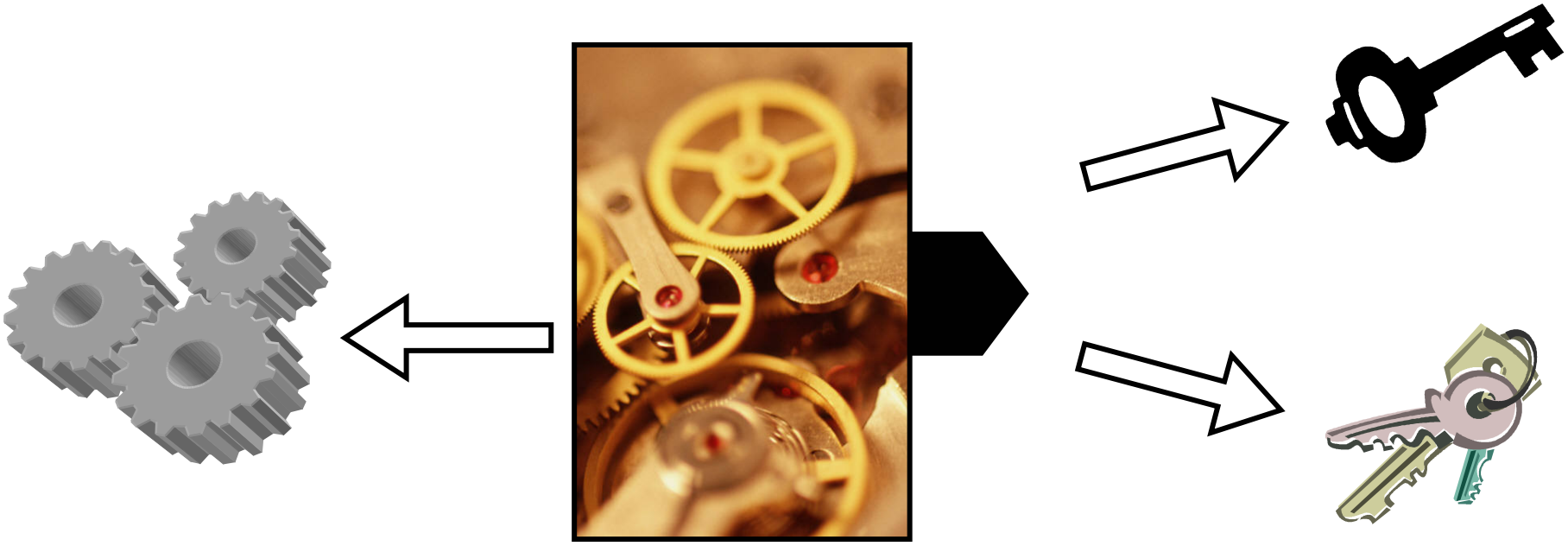
A successful attack

- A successful attack produces a new (tampered) version of “Module” such that
  - It runs on a tampered application
  - It produces the correct sequence of tags
  - The remote server entrusts a tampered client



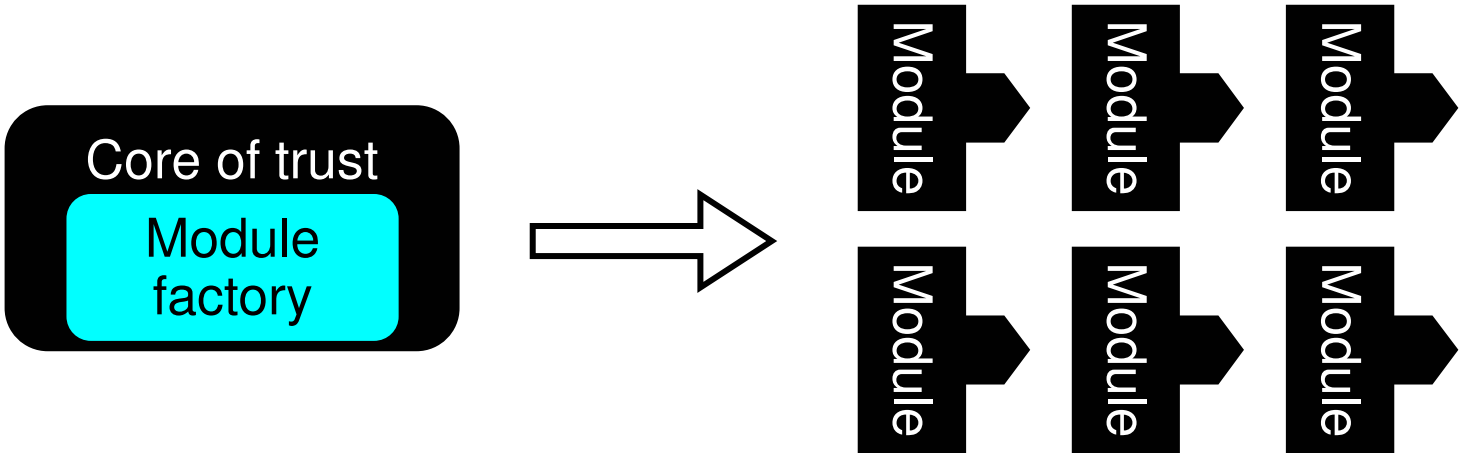
Module weak points

- In the Module, it could be possible to locate data used by, and functionalities devoted to
  - retrieve the currently executed piece of code
  - calculate the authentication code based on a secret key
  - compare the computed authentication code with the expected one
  - produce the tags
  - Insert the tags into the outgoing messages



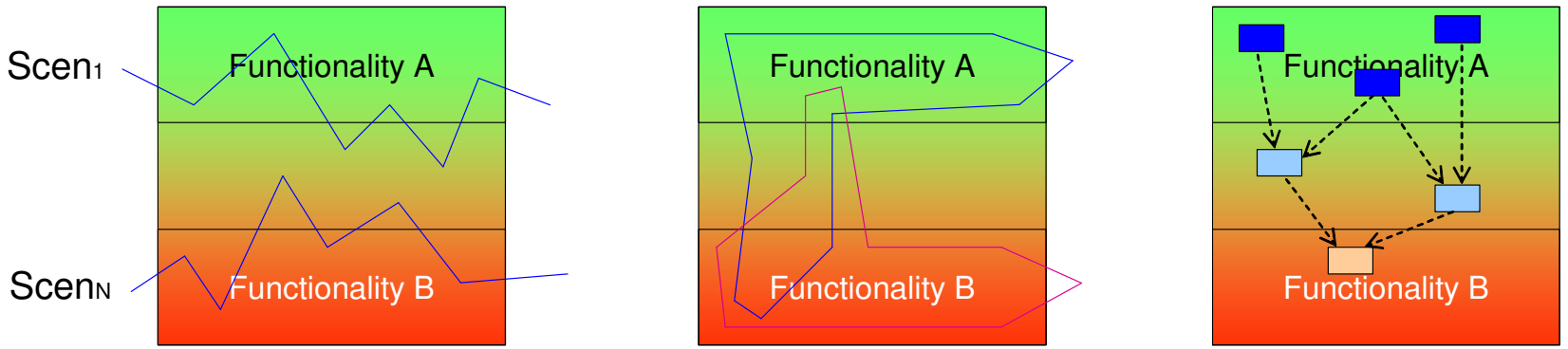
Module factory weak points

- How much information extracted from expired modules can be used to attack a fresh module?
- Is there a repetitive pattern in the module generation mechanism ?
- Possible flaws (weak points) in the generation of modules may suggest improvements



Reverse engineering attacks

- State-of-the-art reverse engineering techniques can be used to locate core functionalities in the Module
  - Feature location
  - Slicing
  - Impact analysis
- These are not fully automated techniques, they require human intervention (how much?)



Reverse engineering complexity

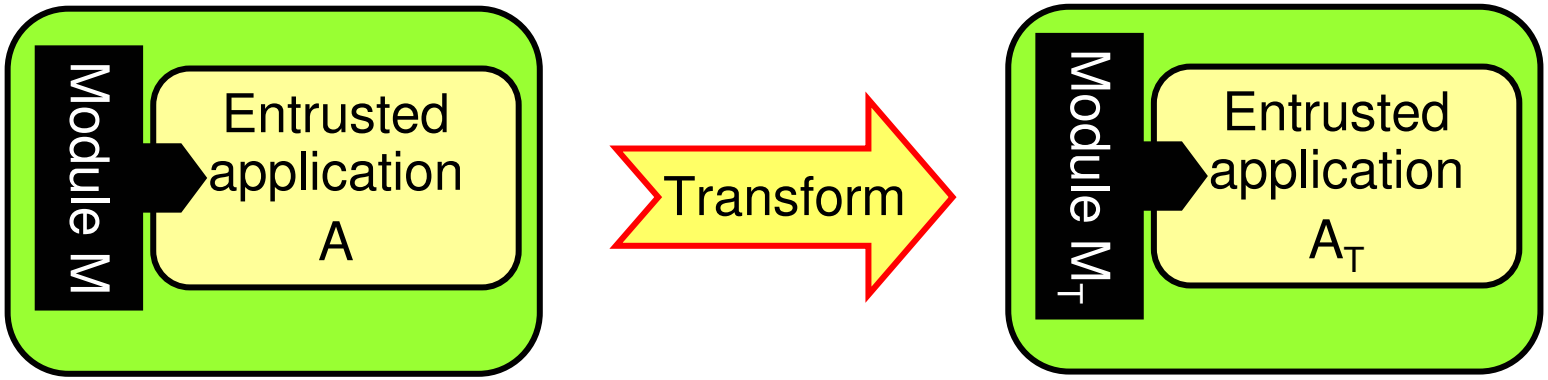
- Metrics of reverse engineering complexity
  - They could be based on the cohesion/coupling between
    - base functionalities and
    - authentication functionalities
  - How much they are separated, how easy is for Reverse Engineering to locate them

Human intervention

- Reverse engineering attacks are not fully automatic.
- Reference scenario comprehend a periodic module replacement
- Reverse engineering attacks have a limited time slot to succeed
- The more human intervention is required the less effective is the attack

Program transformation

- Supposing that all the required data are available
  - How difficult is to tamper with the application and the module?
  - Can it be automated? to which extent?



Conclusion

- There is a general view about the threat model, but details must be investigated
- Metrics/indicator could be used to estimate reverse engineering complexity
- Automated program transformation to tamper with the client application