



Dynamic Secure Aspect Modeling with UML: From Models to Code

Jan Jürjens and Siv Houmb

Software & Systems Engineering
 Technical Univ. of Munich
<http://www4.in.tum.de/~juerjens>

juerjens@in.tum.de


Dep. of Comp. & Inform. Science
 Norwegian Univ. of Science and Technology
siv.hilde.houmb@idi.ntnu.no



Software Engineering & Security


„Penetrate-and-patch“:

- insecure
- disruptive



Traditional formal methods:
 limited adoption in industry.


- training people
- constructing formal specifications.




Jan Jürjens, Siv Houmb: Dynamic Secure Aspect Modeling with UML 2

Security by Design

Increase security with bounded investment in time, costs:



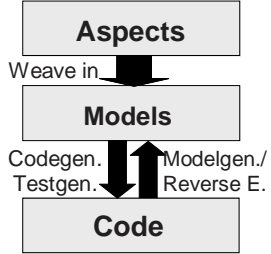
- Weave in security aspects/concerns into artefacts arising in industrial development and use of security-critical systems (UML models, source code, configuration data).
- Tool-supported, theoretically sound, efficient automated security synthesis.




Jan Jürjens, Siv Houmb: Dynamic Secure Aspect Modeling with UML 3

Model-based Security with Aspects

- Weave in security aspects into UMLsec models.
- Generate code (or tests) from models.
- Generate models from evolving or legacy code.





Jan Jürjens, Siv Houmb: Dynamic Secure Aspect Modeling with UML 4


Weaving: Models or Code ?

Weave aspects into model; generate code:
 + admits automated analysis on model-level to validate weaving

Weave aspects into generated code:
 + may be more flexible

Our approach supports both; we prefer (and present here) the first where possible.


Note: woven models not necessarily meant for human consumption.



Jan Jürjens, Siv Houmb: Dynamic Secure Aspect Modeling with UML 5

Model-based Security Aspects

- Define abstract security aspect.
- Define concretization (e.g. protocol).
- If possible, give conditions under which it is secure to weave in aspect using concretization, e.g. by simulation argument.



Jan Jürjens, Siv Houmb: Dynamic Secure Aspect Modeling with UML 6

Secure Channel Aspect

Primary model with directives for security aspects (cf. join points in AspectJ).

To keep d secret, must be sent encrypted.

TUM Jan Jürjens, Siv Houmb: Dynamic Secure Aspect Modeling with UML 7

Secure Channel Aspect: Weaving

Exchange certificate and send encrypted data over Internet.

TUM Jan Jürjens, Siv Houmb: Dynamic Secure Aspect Modeling with UML 8

Aspect Validation

Need to prove concretization securely refines abstract aspect. Challenging problem in security.

For secure channel, have generic result. Often not possible.

→ Use translation validation on the weaving transformation, before or after code generation.

TUM Jan Jürjens, Siv Houmb: Dynamic Secure Aspect Modeling with UML 9

Translation Validation: Model or Code ?

Model:

- + earlier (less work may have to be redone)
- + more abstract → more efficient
- more abstract → may miss attacks
- code construction not completely automatic
- code generators not formally verified

Code:

- + „the real thing“ (which is executed)

→ Do both ! (as far as feasible; e.g. where largely automatic). Here: look at code.

TUM Jan Jürjens, Siv Houmb: Dynamic Secure Aspect Modeling with UML 10

Code-level Translation Validation

Logic-based program understanding of crypto protocols in C which is as

- automatic and
- complete

as possible.

Note: can't be both perfectly automated and complete: Security in general undecidable. Abstract and approximate safely.

TUM Jan Jürjens, Siv Houmb: Dynamic Secure Aspect Modeling with UML 11

Security Analysis in First-order Logic

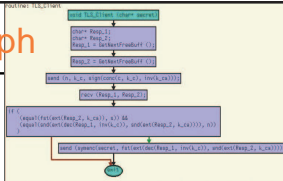
Approximate set of possible data values flowing through system from above.

Predicate $knows(E)$ meaning that the adversary may get to know E during the execution of the protocol.

E.g. secrecy: For any secret s , check whether can derive $knows(s)$ using automated theorem prover.

TUM Jan Jürjens, Siv Houmb: Dynamic Secure Aspect Modeling with UML 12

Control Flow Graph



Generate **control flow graph** (e.g. with aicall (Absint)).

Transform to **state machine**:
 trans(state,inpattern,condition,action,nextstate)
 where action can be outpattern or localvar:=value.

TUM Jan Jürjens, Siv Houmb: Dynamic Secure Aspect Modeling with UML 13

... Translate to First Order Logic

Graph transition
 $TR1 = (in(msg_in), cond(msg_in), out(msg_out))$
 followed by $TR2$ gives predicate $PRED(TR1) =$
 $\forall msg_in. [knows(msg_in) \wedge cond(msg_in) \Rightarrow knows(msg_out) \wedge PRED(TR2)]$

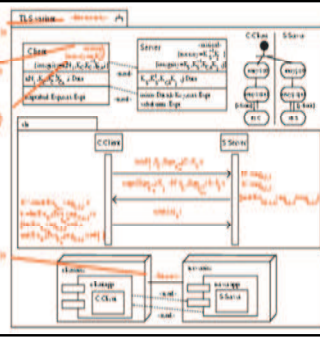
Abstraction (e.g. from senders, receivers): find all attacks, may have false positives.
 Analyze with automated prover.

TUM Jan Jürjens, Siv Houmb: Dynamic Secure Aspect Modeling with UML 14

Example: Proposed Variant of TLS (SSL)

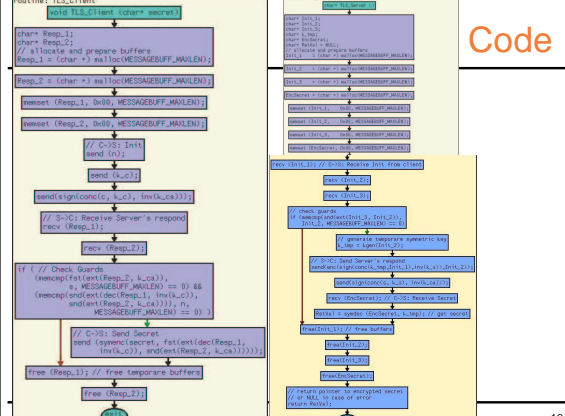
Presented at IEEE Infocom 1999.

Goal: send secret protected by session key using fewer server resources.



TUM Jan Jürjens, Siv Houmb: Dynamic Secure Aspect Modeling with UML 15

Code



TUM Jan Jürjens, Siv Houmb: Dynamic Secure Aspect Modeling with UML 16

Logic

Check whether can derive $knows(s)$.

Surprise: Yes!

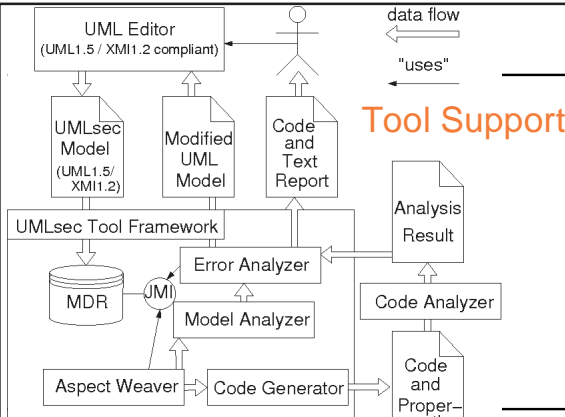
That means: Aspect concretization does **not** preserve secrecy of s → not secure.

```

input_formula(tls_abstract_protocol_axiom, (
  !([ArgS_11, ArgS_12, ArgS_13, ArgC_11, ArgC_12] : (
    [DataC_KK, DataC_K, DataC_n] : (
      % Client -> Attacker (1. message)
      (
        knows(a)
        & knows(k,c)
        & knows(sign(conc(c, k_c)))
      )
      % Server -> Attacker (2. ...)
      (
        knows(Args_11)
        & knows(Args_c)
        & knows(Args_n)
        & (? P ...)
      )
    )
  )
)
    
```

TUM Jan Jürjens, Siv Houmb: Dynamic Secure Aspect Modeling with UML 17

Tool Support



TUM Jan Jürjens, Siv Houmb: Dynamic Secure Aspect Modeling with UML 18

Industrial Application

Biometric Authentication System in development by large German company in joint project.



Use our design and validation methods to develop and analyze system.

Discovered **three significant security weaknesses** against subsequently improved versions.



Related Work

- R. France, G. Georg et al.: use security aspect models to describe crosscutting solutions.
- H. Gomaa et al.: Separate application and security concerns in modeling.

So far mostly emphasis on static models. Also so far no integration with code-level verification.



Conclusions, Future Work

Aspect-based Security with UMLsec:

- **formally based** approach
- sophisticated, **automated tool** support
- successful use in **industrial projects**



Further work: Link to formal semantics in development in joint project with B. Selic (IBM Rational) et al.; integrate into NSF initiative by R. France et al..

More information (papers, slides, tool, ...):
<http://www.umlsec.org>

