Formal Development and Verification of Security-Critical Systems with UML

Jan Jürjens

Computing Laboratory, University of Oxford

jan@comlab.ox.ac.uk

http://www.jurjens.de/jan
Challenges in formal development/verification of secure systems:

- formal specification usually unavailable (and expensive)
- only small security-critical parts are feasible
- technical problems: composition, refinement
- only feasible to give simplified account
- vulnerabilities from bugs in implementation
Towards solutions

- use (formal core of) Unified Modeling Language (UML)
- diagrams give different views (context, physical layer)
- security notions composable, preserved under refinement
- soundness of symbolic reasoning wrt. complexity-theory
- specification-based testing
Security vs. Refinement

Goal: develop secure systems by stepwise refinement from abstract specifications to concrete specifications.

Problem: common formulations of security properties are not preserved by refinement ("refinement paradox")!

Applies in particular to implementations (usually refinements of specifications)!!
UMLsec (fragment)

- **Activity diagrams**: secure control flow, coordination
- **Statechart diagram**: security preserved within object
- **Class diagram**: exchange of data preserves security levels
- **Interaction diagram**: security-critical interaction
- **Deployment diagram**: physical security requirements
Distributed Objects

Objects distributed over untrusted networks.

“Adversary” intercepts, modifies, deletes, inserts messages.

Cryptography provides security.
Expressions

\[ E ::= \]
\[
\begin{align*}
    &d & \quad & d \in D \\
    &K & \quad & \text{key (} K \in \text{Keys}) \\
    &x & \quad & x \in \text{Var} \\
    &E_1 :: E_2 & \quad & \text{concatenation} \\
    &\{E\}_e & \quad & \text{encryption} \ (e \in \text{Keys} \cup \text{Var}) \\
    &\text{Dec}_e(E) & \quad & \text{decryption} \ (e \in \text{Keys} \cup \text{Var})
\end{align*}
\]

\( K^{-1} \): decryption key corresponding to encryption key \( K \).

Postulate \( \text{Dec}_{K^{-1}}(\{E\}_K) = E \).
Statechart: \( S = (S, i, T) \) where

- \( S \) set of (simple) states
- \( i \) initial state
- \( T \) set of (external) transitions \((source, target, event, guard, action)\),
  \( event, action \) are messages of form \( op(exp_1, \ldots, exp_n)\),
  \( guard \) propositional expression
Statecharts: Semantics

\( \text{Msg}_X \): set of sequences of messages \( \text{op}(\exp_1, \ldots, \exp_n) \) with \( \text{op} \in X \).

Statechart \( S \) defines function \( \llbracket S \rrbracket : \text{Msg}_I \to \mathcal{P}(\text{Msg}_O) \)
defined inductively for \( s \in S \) by \( \llbracket s \rrbracket(\epsilon) \overset{\text{def}}{=} \epsilon \) and:

\[
\llbracket s \rrbracket(\text{op}_1(\vec{b}).\text{events}) \overset{\text{def}}{=} \bigcup_{t,s',\vec{a}_2'} \text{op}_2(\vec{a}_2').\llbracket s' \rrbracket(\text{events})
\]

with \( s \xrightarrow{t} s' \), where \( t = (\text{op}_1(\vec{a}_1), \text{guard}, \text{op}_2(\vec{a}_2)) \), 
\( \text{guard}[\vec{a}_1/\vec{b}] \) holds and \( \vec{a}_2(\vec{b}) = \vec{a}_2' \), if such exist.

\[
\llbracket s \rrbracket(\text{op}_1(\vec{b}).\text{events}) \overset{\text{def}}{=} \llbracket s \rrbracket(\text{events}), \text{ otherwise.}
\]

Then \( \llbracket S \rrbracket \overset{\text{def}}{=} \llbracket i \rrbracket \) for initial state \( i \).
Composition

For $f_i : \text{Msg}_{I_i} \to \mathcal{P}(\text{Msg}_{O_i})$ $(i = 1, 2)$ with $O_1 \cap O_2 = \emptyset$:

Define $f_1 \otimes f_2 : \text{Msg}_I \to \mathcal{P}(\text{Msg}_O)$

by $f_1 \otimes f_2(\vec{s}) = \{\vec{t} | O : \vec{t} | I = \vec{s} | I \land \vec{t} | O_i \in f_i(\vec{s} | I_i)(i = 1, 2)\}$

(where $\vec{t} \in \text{Msg}_{I \cup O}$, $I = (I_1 \cup I_2) \setminus (O_1 \cup O_2)$ and

$O = (O_1 \cup O_2) \setminus (I_1 \cup I_2)$).
Secrecy

$f$ may eventually output $E$ if

exists input Msgs $\vec{s}$, output Msgs $\vec{t} \in f(\vec{s})$, such that $E$ appears in $\vec{t}$.

**Definition** $P$ preserves the secrecy of $m \in \text{Keys}$ if

exists no $A$ such that $[P] \otimes [A]$ may eventually output $m$ (and $m \not\in K_A$).

Protects atomic values (following Dolev, Yao 1983).

$\{m\}_K :: K$ does not preserve secrecy of $m$ or $K$.
$\{m\}_K$ does.
Refinement

**Definition** $Q$ refines $P$ ($P \leadsto Q$) if for each $\vec{s} \in \text{Msg}_{I_P}$ have $[P](\vec{s}) \supseteq [Q](\vec{s})$.

**Theorem**
- If $P$ preserves secrecy of $m$ and $P \leadsto Q$
  then $Q$ preserves secrecy of $m$. 
Why might this be true?

Separate two kinds of non-determinism:

- underspecification
- unpredictability (key generation)
Related Work

Formal semantics for UML (Evans, France, Lano, Rumpe 98; Bolton, Davis; Crichton; Cavarra)

Formal verification of security protocols
(Burrows, Abadi, Needham; Roscoe, Lowe, …; FME 01, FASE 01)


S. Schneider (1996): Confidentiality property preserved under refinement. No cryptographic primitives considered.
Conclusion

Formal development of security-critical systems with formal core of UML;

secrecy preserved by refinement.
Further Work

Compositionality (MMM 01)

Common Electronic Purse Specifications (Ifip SEC 01)

Encapsulating security engineering knowledge (IWSecP 01)

Specification-based testing (PSI 01)

Java Security

Extension of UML using profiles
Future Work

More aspects of security.

Relate different views.

Tool support.