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

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

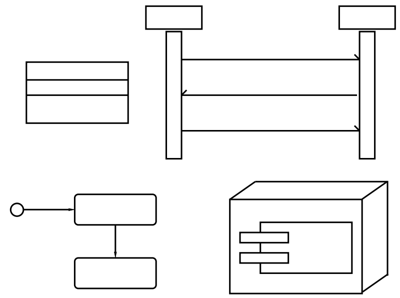
from abstract specifications to concrete specifications.

Problem: common formulations of security properties are

not preserved by refinement ("refinement paradox") !

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

UMLsec (fragment)



- Acivity 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.

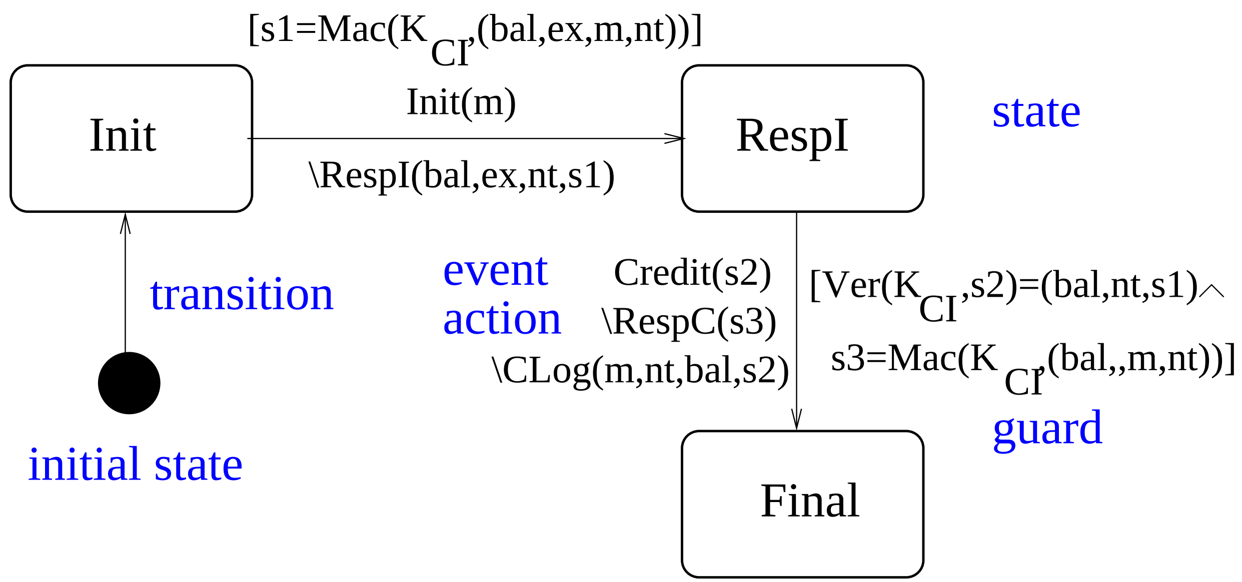
Expressions

$E ::=$	expression
$\quad d$	$d \in \mathcal{D}$
$\quad K$	key ($K \in \mathbf{Keys}$)
$\quad x$	$x \in \mathbf{Var}$
$\quad E_1 :: E_2$	concatenation
$\quad \{E\}_e$	encryption ($e \in \mathbf{Keys} \cup \mathbf{Var}$)
$\quad \mathcal{D}ec_e(E)$	decryption ($e \in \mathbf{Keys} \cup \mathbf{Var}$)

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

Postulate $\mathcal{D}ec_{K^{-1}}(\{E\}_K) = E$.

Statechart diagrams



Statechart: $\mathcal{S} = (S, i, T)$ 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 $\mathsf{op}(\mathsf{exp}_1, \ldots, \mathsf{exp}_n)$,

  $guard$ propositional expression

Statecharts: Semantics

Msg_X: set of sequences of messages $\mathsf{op}(\mathsf{exp}_1, \ldots, \mathsf{exp}_\mathsf{n})$ with $\mathsf{op} \in X$.

Statechart $\mathcal{S}$ defines function $[\![\mathcal{S}]\!] : \mathsf{Msg}_I \to \mathcal{P}(\mathsf{Msg}_O)$
defined inductively for $s \in S$ by $[\![s]\!](\varepsilon) \stackrel{\text{def}}{=} \varepsilon$ and:

$$[\![s]\!](op_1(\vec{b}).events) \stackrel{\text{def}}{=} \bigcup_{t, s', \vec{a}'_2} op_2(\vec{a}'_2).[\![s']\!](events)$$

with $s \xrightarrow{t} s'$, where $t = (op_1(\vec{a}_1), guard, op_2(\vec{a}_2))$,
$guard[\vec{a}_1/\vec{b}]$ holds and $\vec{a}_2(\vec{b}) = \vec{a}'_2$, if such exist.

$[\![s]\!](op_1(\vec{b}).events) \stackrel{\text{def}}{=} [\![s]\!](events)$, otherwise.

Then $[\![\mathcal{S}]\!] \stackrel{\text{def}}{=} [\![i]\!]$ for initial state i.

Composition

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

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

by $f_1 \otimes f_2(\vec{s}) = \{\vec{t}\downharpoonright_O : \vec{t}\downharpoonright_I = \vec{s}\downharpoonright_I \wedge \vec{t}\downharpoonright_{O_i} \in f_i(\vec{s}\downharpoonright_{I_i})(i = 1, 2)\}$

(where $\vec{t} \in \mathsf{Msg}_{I \cup O}$, $I = (I_1 \cup I_2) \setminus (O_1 \cup O_2)$ 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 $m \in \mathbf{Keys}$ if
exists no A such that $[\![P]\!] \otimes [\![A]\!]$ may eventually output m
(and $m \notin K_A$).

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 ($P \rightsquigarrow Q$) if for each $\vec{s} \in \mathsf{Msg}_{I_P}$ have $[\![P]\!](\vec{s}) \supseteq [\![Q]\!](\vec{s})$.

Theorem

- If P preserves secrecy of m and $P \rightsquigarrow Q$ 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)

McLean (1996): Possibilistic security properties not in Alpern/Schneider framework.

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.