Information flow control revisited:

Noninfluence = Noninterference + Nonleakage

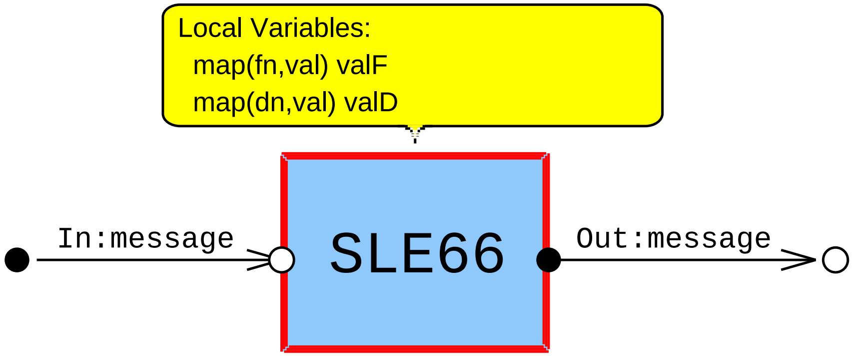
David von Oheimb

David.von.Oheimb@siemens.com

Siemens, Corporate Technology, D-81730 Munich

Motivation

Task: Security analysis for Infineon SLE66 smart card processor



Main concern: confidentiality of on-chip secrets

Initial solution: secret values do not appear as output

Problem: leakage of re-encoded and partial information

***Maximal* solution:** observable output *independent* of secrets

Approach: some sort of noninterference

Overview

- noninterference
  - classical notion
  - unwinding
  - nondeterminism
  - improvements
- nonleakage
  - motivation, notion, variants
  - application
- noninfluence
- insights

Generic notions

System model: — Moore automaton

$step : action \times state \to state$

$run : action^* \times state \to state$

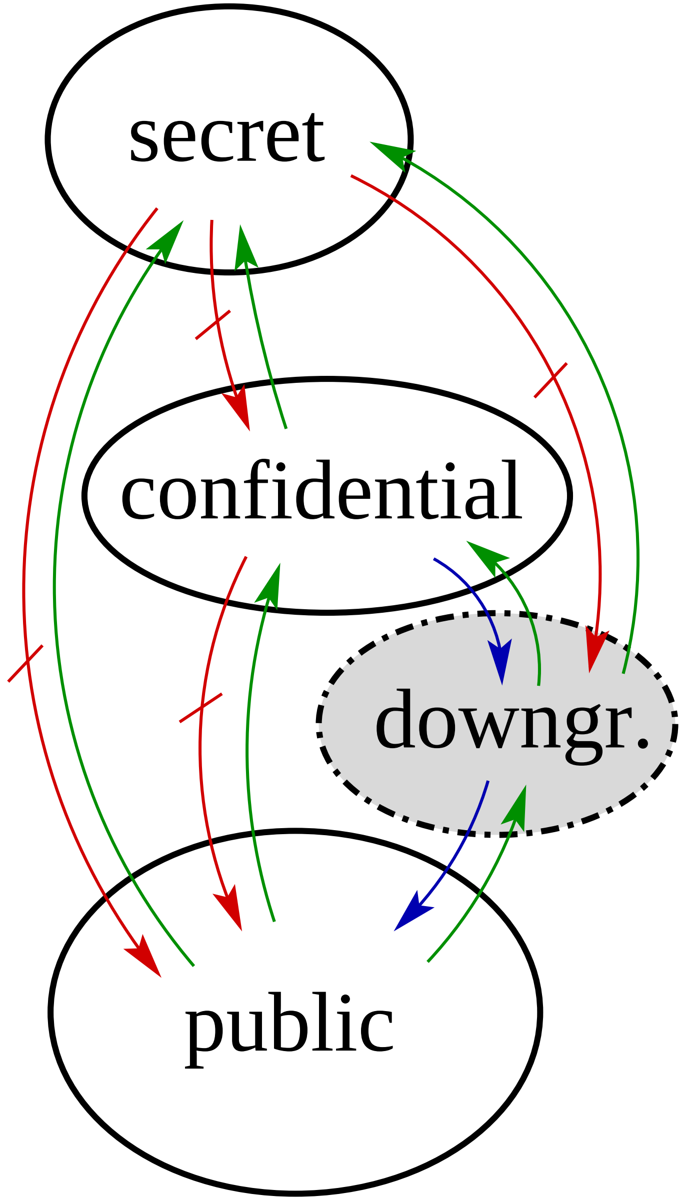
— also nondeterministic variants

Security model:

$domain$ — secrecy level/area

$obs : domain \times state \to output$

$dom : action \to domain$ — input domain



Policy or interference relation

$\rightsquigarrow : \wp(domain \times domain)$

— always reflexive, possibly intransitive

Noninterference relation: $\not\rightsquigarrow$

Noninterference [GM82/84,Rus92]

Aim: secrecy of the presence/absence of actions

$$noninterference \equiv$$
$$\forall \alpha\ u.\ obs(u, run(\alpha, s_0)) = obs(u, run(ipurge(u, \alpha), s_0))$$

$ipurge(u, \alpha) =$"remove from the sequence α all actions that may not influence u, directly or via the domains of subsequent actions within α"

Observational equivalence/relation

$$\cdot \lhd \cdot \overset{\cdot}{\simeq} \cdot \lhd \cdot :\ domain\ \rightarrow \wp(state\ \times\ action^*\ \times\ state\ \times\ action^*)$$
$$s \lhd \alpha \overset{u}{\simeq} t \lhd \beta \equiv obs(u, run(\alpha, s)) = obs(u, run(\beta, t))$$

$$noninterference \equiv \forall \alpha\ u.\ s_0 \lhd \alpha \overset{u}{\simeq} s_0 \lhd ipurge(u, \alpha)$$

ipurge & sources

$$ipurge : domain \times action^* \rightarrow action^*$$
$$ipurge(u, []) = []$$
$$ipurge(u, a \frown \alpha) = if\ dom(a) \in sources(a \frown \alpha, u)$$
$$then\ a \frown ipurge(u, \alpha)\ else\ ipurge(u, \alpha)$$

$sources(\alpha, u)$ = “all domains of actions in α that may influence u, directly or via the domains of subsequent actions within α”

e.g., $v \in sources(a_1 \frown a_2 \frown a_3 \frown a_4, u)$
if $v = dom(a_2) \rightsquigarrow dom(a_4) \rightsquigarrow u$ (even if $v \not\rightsquigarrow u$)

$$sources : action^* \times domain \rightarrow \wp(domain)$$
$$sources([], u) = \{u\}$$
$$sources(a \frown \alpha, u) = sources(\alpha, u) \cup$$
$$\{w.\ \exists v.\ dom(a) = w \land w \rightsquigarrow v \land v \in sources(\alpha, u)\}$$

Unwinding

Problem: noninterference is global property, to be shown for any α

Idea: induction on α shows preservation of

unwinding relation $\sim : domain \rightarrow \wp(state \times state)$

— some kind of equality on the sub-state belonging to the domain

— no need to be reflexive, symmetric, nor transitive [Man00/03]

— lifting to sets of domains: $s \overset{U}{\approx} t \equiv \forall u \in U.\ s \overset{u}{\sim} t$

Local properties: essentially $s \overset{u}{\sim} t \longrightarrow step(a,s) \overset{u}{\sim} step(a,t)$

(step consistency, step respect, local respect)

Proof sketch

Theorem Goal: $obs(u, run(\alpha, s_0)) = obs(u, run(ipurge(u, \alpha), s_0))$

Main Lemma: $\forall s\ t.\ s \overset{sources(\alpha,u)}{\approx} t \longrightarrow run(\alpha, s) \overset{u}{\sim} run(ipurge(u, \alpha), t)$

Proof of Theorem: specialize by $s = t = s_0$, use $s_0 \overset{sources(\alpha,u)}{\approx} s_0$,

and apply output consistency $\forall u\ s\ t.\ s \overset{u}{\sim} t \longrightarrow obs(u, s) = obs(u, t)$

Proof of Main Lemma: by induction $\alpha' \longrightarrow a \frown \alpha'$

$s \overset{sources(a \frown \alpha',u)}{\approx} t$ implies

$if\ dom(a) \in sources(a \frown \alpha', u)$

(step consistency + respect): $then\ step(a, s) \overset{sources(\alpha',u)}{\approx} step(a, t)$

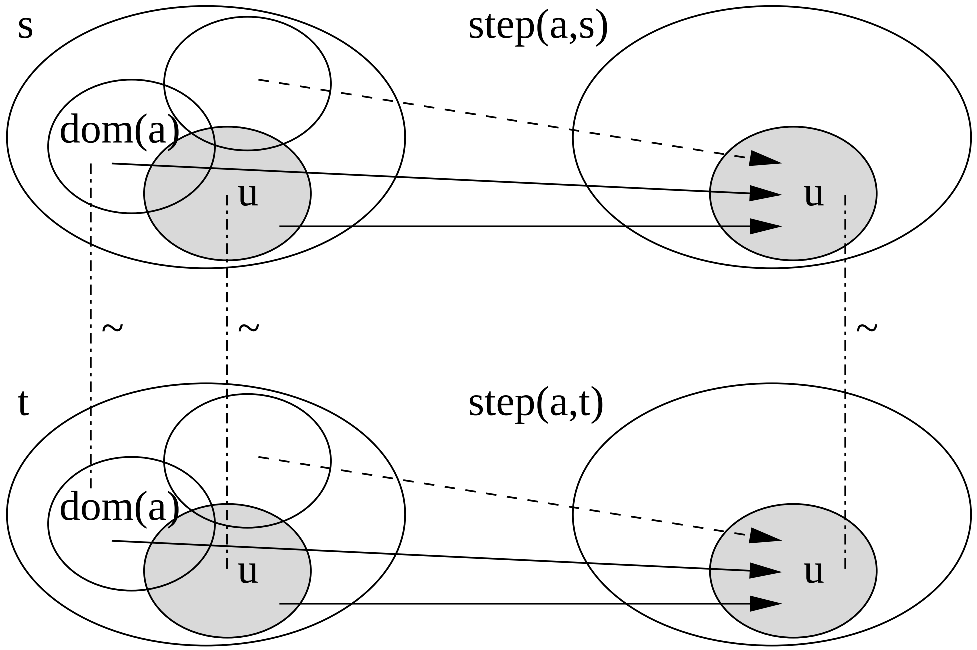
(local respect): $else\ step(a, s) \overset{sources(\alpha',u)}{\approx} t$, then

ind. hypothesis implies $run(\alpha', step(a, s)) \overset{u}{\sim} run(ipurge(u, a \frown \alpha'), t)$

Step consistency and respect

$weakly\_step\_consistent \equiv$

$$\forall a\ u\ s\ t.\ dom(a) \rightsquigarrow u \ \wedge\ s \overset{dom(a)}{\sim} t \ \wedge\ s \overset{u}{\sim} t \longrightarrow step(a,s) \overset{u}{\sim} step(a,t)$$



$$step\_respect \equiv \forall a\ u\ s\ t.\ dom(a) \not\rightsquigarrow u \ \wedge\ s \overset{u}{\sim} t \longrightarrow step(a,s) \overset{u}{\sim} step(a,t)$$

$$local\_respect\_left \equiv \forall a\ u\ s\ t.\ dom(a) \not\rightsquigarrow u \ \wedge\ s \overset{u}{\sim} t \longrightarrow step(a,s) \overset{u}{\sim} t$$

$$local\_respect\_right \equiv \forall a\ u\ s\ t.\ dom(a) \not\rightsquigarrow u \ \wedge\ s \overset{u}{\sim} t \longrightarrow s \overset{u}{\sim} step(a,t)$$

Nondeterminism

$Step: \ action \ \rightarrow \ \wp(state \ \times \ state)$ new: non-unique outcome,

$Run: \ action^* \ \rightarrow \ \wp(state \ \times \ state)$ partiality/reachability

$$Noninterference \equiv \forall \alpha\ u\ \beta.\ ipurge(u,\alpha) = ipurge(u,\beta) \longrightarrow$$
$$\forall s.\ (s_0, s) \in Run(\alpha) \longrightarrow \exists t.\ (s_0, t) \in Run(\beta) \wedge obs(u,s) = obs(u,t)$$

Complications for weak step consistency $\Rightarrow$
stronger notions preserving simultaneous unwinding relation $\approx$:
uniform step consistency, step respect, and (right-hand) local respect

Requires in general more proof effort, yet not for two important cases:

- functional $Step(a)$
- two-level domain hierarchy $\{H, L\}$

Improvements over [Rus92]

- weak step consistency suffices also for transitive policies
- improvements on access control interpretation:
  - transitivity of policy not required
  - observable locations need not form a hierarchy
  - stronger preconditions of 2^{nd} reference monitor assumption
- no restrictions on unwinding relation
- extension to nondeterminism

Nonleakage and Noninfluence

Event-based systems:

- visibility of actions/events is primary,
- secret state is secondary (via side-effects)

$\Rightarrow$ Noninterference

State-oriented systems:

- secret state is primary,
- actions/events are secondary or irrelevant

$\Rightarrow$ Nonleakage

State-event-systems:

- visibility of actions/events is relevant
- also secrecy in state is essential

$\Rightarrow$ Noninfluence

Concept

Language-based security: no assignments of high-values to low-variables, enforced by type system

Semantically: take (x, y) as elements of the state space with high-level data (on left) and low-level data (on right).

Step function $S(x, y) = (S_H(x, y), S_L(x, y))$ does not leak information from high to low if $S_L(x_1, y) = S_L(x_2, y)$ (functional independence).

Observational equivalence $(x, y) \overset{L}{\sim} (x', y') \;:\longleftrightarrow\; y = y'$ allows re-formulation:

$s \overset{L}{\sim} t \longrightarrow S(s) \overset{L}{\sim} S(t)$ (preservation of $\overset{L}{\sim}$)
step consistency + respect

Generalization to action sequences α and arbitrary policies $\rightsquigarrow$

Definition

$$nonleakage \equiv \forall \alpha\ s\ u\ t.\ s \overset{sources(\alpha,u)}{\approx} t \longrightarrow s \lhd \alpha \overset{u}{\simeq} t \lhd \alpha$$

"the outcome of u's observation is independent of those domains from which no (direct or indirect) information flow is allowed."

- like Main Lemma, but no purging (visibility of actions irrelevant)
- unwinding relation $\sim$ is part of the notion:
  the secrets for u are those state components not constrained by $\sim$
- corresponding unwinding theorem: nonleakage implied by

  $weakly\_step\_consistent \ \wedge \ step\_respect \ \wedge \ output\_consistent$

Variants

If (domains of) actions are irrelevant:

$$weak\_nonleakage \equiv \forall \alpha\ s\ u\ t.\ s \overset{chain(\alpha,u)}{\approx} t \longrightarrow s \lhd \alpha \overset{u}{\simeq} t \lhd \alpha$$

where $chain : action^* \times domain \rightarrow \wp(domain)$

e.g., $v \in chain(a_1 \frown a_2 \frown a_3 \frown a_4, u)$ if $\exists v'.\ v \leadsto v' \leadsto u$

- implied by $output\_consistent \wedge weak\_step\_consistent\_respect$

Weak combination of step consistency and step respect:

$$\forall s\ u\ t.\ s \overset{\{w.\ w \leadsto u\}}{\approx} t \longrightarrow \forall a.\ step(a,s) \overset{u}{\sim} step(a,t)$$

If additionally the policy is transitive:

$$trans\_weak\_nonleakage \equiv \forall s\ u\ t.\ s \overset{\{w.\ w \leadsto u\}}{\approx} t \longrightarrow \forall \alpha.\ s \lhd \alpha \overset{u}{\simeq} t \lhd \alpha$$

- implied by $weak\_step\_consistent\_respect \wedge output\_consistent$

Infineon SLE66 Case Study

Security objective: secret functionality and data is not leaked

Applied notion: nondeterministic transitive weak Nonleakage

Unwinding: equality on: non-secrets, phase, function availability

Minor complication: invariants required ($\Rightarrow$ reachable states)

Results:

- underspecified functions require nonleakage assumptions
- anticipated (non-critical) single data leakage confirmed
- availability of secret functions is leaked
  $\rightsquigarrow$ security objectives clarified: availability is public
- no other information leaked

Noninfluence

combining noninterference and nonleakage:

$$noninfluence \equiv \forall \alpha\ s\ u\ t.\ s \overset{sources(\alpha,u)}{\approx} t \longrightarrow s \lhd \alpha \overset{u}{\simeq} t \lhd ipurge(\alpha, u)$$

- useful if both . . .
  - certain actions should be kept secret and
  - initially present secret data should not leak
- stronger than $noninterference$
- implied by
  $weakly\_step\_consistent \wedge local\_respect \wedge output\_consistent$
- appeared already as Main Lemma (Rushby’s Lemma 5)

Insights on observations and unwinding

- for given α, observational equivalence
  $s \lhd \alpha \overset{u}{\simeq} t \lhd \alpha \equiv obs(u, run(\alpha, s)) = obs(u, run(\alpha, t))$
  as equal outcome of tests on s and t by u executing α
- observational preorder $s \lhd \alpha \overset{u}{\precsim} t \lhd \alpha \equiv \forall s'.\ (s, s') \in Run(\alpha) \longrightarrow \exists t'.\ (t, t') \in Run(\alpha) \wedge obs(u, s') = obs(u, t')$
  also entails preservation of enabledness of α
- observation may be encoded by enabledness of “probing” actions
  $\Rightarrow$ observational preorder $\simeq$ enabledness relation [Mantel-PhD03]
- observational equivalence/preorder induced by obs …
  - is reflexive and transitive
  - is implied by unwinding (which maybe is not an equivalence)
  - can be regarded as reflexive+transitive closure of unwinding
  - often coincides with unwinding relation

Conclusion

- refinements and generalizations on Rushby's work
- introduction of new notions for data flow security:
  noninterference + nonleakage = noninfluence
- insights on unwinding and observation relations
- application in machine-assisted security analysis:
  - smart card processors (secrecy)
  - operating systems (process separation)