Composability of Secrecy*

Jan Jürjens

Computing Laboratory, University of Oxford

www.jurjens.de/jan

*Initial work performed at Bell Labs Research, Lucent Technologies.

Security vs. Modularity

Goal: develop secure systems by composition

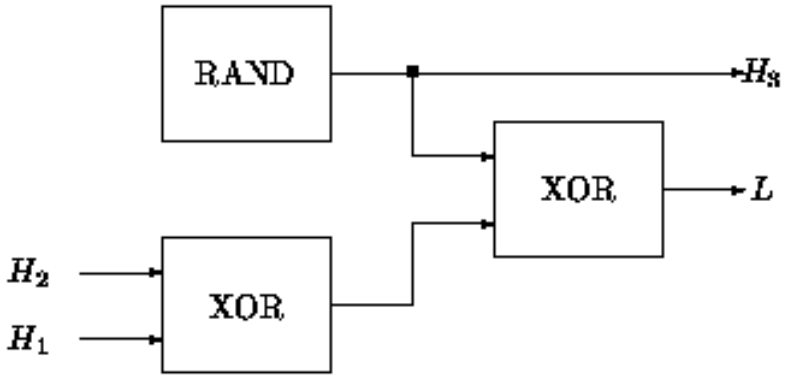
of whole system from components.

Problem: common formulations of security properties are

not preserved by composition !

Applies in particular to implementations
(to be executed with other systems) !!

Example (1)



May seem to keep the value of H_3 secret.

But: Not so if H_3 fed back into H_2 (without delay) !

Other examples: secure information flow models by McLean and Gray.

Example (2)

$p ::= K$ and $p ::= \{m\}_K$

Each may seem to keep m secret,

but not their composition.

This Work

- Use standard specification language with standard notion of composition.
- Give conditions under which standard secrecy property is preserved by composition.
- Also preserved under refinement.

Specification language

Focus (M. Broy):

- Synchronously executing processes modelled by stream-processing functions.
- Interaction: transmission of data over unidirectional FIFO channels.
- Communication asynchronous (no refusal by receiver).

Extension: specification language with cryptographic primitives.

Specification Language: Expressions

Exp: empty expression ε and the non-empty expressions:

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

K^{-1}: decryption key corresponding to encryption key K.
Assume $\mathcal{D}ec_{K^{-1}}(\{E\}_K) = E$.

Programs

$p ::=$	program
$\quad E$	output expression
$\quad either\ p\ or\ q$	nondeterministic branching
$\quad if\ E = E'\ then\ p\ else\ q$	conditional
$\quad case\ E\ of\ key\ do\ p\ else\ q$	determine whether key
$\quad case\ E\ of\ x :: y\ do\ p\ else\ q$	break up list

Program computes one channel output at time $t + 1$
from input values at time t.

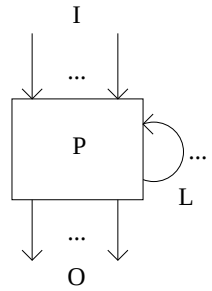
Iteration using local store.

Example $case\ c\ of\ key\ do\ \{d\}_c\ else\ \varepsilon$ outputs value from d
encrypted under value from c if it's a key, otherwise ε.

Processes

A process is of the form $P = (I, O, L, (p_c)_{c \in O \cup L})$ where

- $I \subseteq \mathbf{Channels}$ (input channels)
- $O \subseteq \mathbf{Channels}$ (output channels)
- $L \subseteq \mathbf{Channels}$ (local channels)
- p_c: closed program with input channels in $I \cup L$ and output channel $c \in O \cup L$.



Write $\tilde{I} \stackrel{\text{def}}{=} I \cup L$.

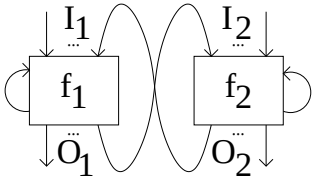
Stream-processing functions

$\mathbf{Stream}_C \stackrel{\text{def}}{=} (\mathbf{CExp}^\omega)^C$: C-indexed tuples of sequences of closed expressions.

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

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

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



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

Interpretation

Each process $P = (I, O, L, (p_c)_{c \in \tilde{O}})$ defines stream-processing function $[\![P]\!] : \mathbf{Stream}_I \to \mathcal{P}(\mathbf{Stream}_O)$.

$[\![P]\!]$ causal: $n + 1^{st}$ output depends only on first n inputs.

Adversaries A may be non-causal: slightly different interpretation $[\![A]\!]_r$ (sometimes rushing adversaries, B. Pfitzmann).

Composition

Composition of $P = (I_P, O_P, L_P, (p_c)_{c \in O_P \cup L_P})$
and $Q = (I_Q, O_Q, L_Q, (p_c)_{c \in O_Q \cup L_Q})$

$$P \otimes Q \stackrel{\text{def}}{=} (I, O, L, (p_c)_{c \in O \cup L})$$

where $I = (I_P \cup I_Q) \setminus (O_P \cup O_Q)$, $O = (O_P \cup O_Q) \setminus (I_P \cup I_Q)$
and $L = L_P \cup L_Q \cup ((I_P \cup I_Q) \cap (O_P \cup O_Q))$.

Secrecy (1)

f may eventually output E if
exists input-stream $\vec{s}$, output stream $\vec{t} \in f(\vec{s})$,
output channel c and point in time i such that $(\vec{t}(c))_i = E$.

Definition P preserves the secrecy of $m \in \mathbf{Secret} \cup \mathbf{Keys}$ if
exists no A such that $[\![P]\!] \otimes [\![A]\!]_r$ may eventually output m
(and $m \notin S_A \cup K_A$).

Protects atomic values (following Dolev, Yao 1983).

$p \stackrel{\text{def}}{=} \{m\}_K :: K$ does not preserve secrecy of m or K.
$p \stackrel{\text{def}}{=} \{m\}_K$ does.

Secrecy (2)

Definition

• P protects the secrecy of m with $\mathcal{K}$ if for any A with $m \notin S_A \cup K_A$ such that $[\![P]\!] \otimes [\![A]\!]_r$ may eventually output m, have $\mathcal{K} \subseteq K_A$.

• P respects the protection of m by $\mathcal{K}$ if for any Q that protects the secrecy of m with $\mathcal{K}$ and any A with $m \notin S_A \cup K_A$ and $\mathcal{K} \not\subseteq K_A$ such that $[\![P]\!] \otimes [\![Q]\!] \otimes [\![A]\!]_r$ may eventually output m,
there is A' with $m \notin S_{A'} \cup K_{A'}$ and $\mathcal{K} \cap K_{A'} = \emptyset$
such that $[\![A']\!]_r \otimes [\![Q]\!] \otimes [\![A]\!]_r$ may eventually output m.

The process that decrypts input with K and outputs it respects the protection of any m with $\{K'\}$ for $K \neq K'$.

Compositionality

Theorem

Suppose that P and P' protect secrecy of m with $\mathcal{K}$
and respect protection of m by $\mathcal{K}$.
Then $P \otimes P'$ protects secrecy of m with $\mathcal{K}$
and respects protection of m by $\mathcal{K}$.

Examples resolved

(1) Computation introduces delay.

(2) Keep track of protecting keys.

Secrecy (intensionally)

Definition P protects m intensionally with $\mathcal{K}$ if
exists set of local channels $H_P \subseteq L_P$ such that
an output p_c to channel c may contain a subexpression

- $\mathcal{D}ec_e(E)$ only if
  - $c \in H_P$, or
  - e is protected from being evaluated to key in $\mathcal{K}$, or
  - E is protected from containing m as subexpression,
- m or d (for $d \in H_P$) only if
  - $c \in H_P$, or
  - it occurs as the key used in an encryption or decryption, or
  - it is encrypted under the keys in $\mathcal{K}$

Soundness of intensional version

Theorem

If P protects m intensionally with $\mathcal{K}$

then P protects m with $\mathcal{K}$.

Refinement

Definition

Define $P \rightsquigarrow P'$ if for each $\vec{s} \in \mathbf{Stream}_{I_P}$, have $[\![P]\!](\vec{s}) \supseteq [\![P']\!](\vec{s})$.

Theorem

For each above secrecy property p have:

If P satisfies p and $P \rightsquigarrow P'$ then P' satisfies p.

Related Work

Varadharajan (1991): hook-up property for information flow secure nets

Meadows (1992): composability using traces based on procedure calls

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

Jürjens (Concur 2000): Secure information flow by McLean and Gray not composable (and composable modification)

Jürjens (FME 2001): Secrecy preserved under refinement

Conclusion

- Secrecy property preserved by composition (under conditions).

- earlier: preserved under refinement; flaw in TLS variant

$\rightsquigarrow$ Theoretically sound notion of secrecy
with practical applicability.

Further Work

Refinement of formal crypto to complexity-theory (with M. Abadi)

Extension to other properties.

Foundation for developing secure systems with (formal core of) UML (FASE/ETAPS'01, Security Protocols '01, IFIP SEC'01,. . .)

www.jurjens.de/jan

Model: Programs

Any closed p defines $[p] : \mathbf{CExp}^{\tilde{I}_p} \to \mathcal{P}(\mathbf{CExp})$:

- $[E](\vec{E}) = \{E(\vec{E})\}$
- $[either\ p\ or\ q](\vec{E}) = [p](\vec{E}) \cup [q](\vec{E})$
- $[if\ E = E'\ then\ p\ else\ q](\vec{E}) = [p](\vec{E})$ if $[E](\vec{E}) = [E'](\vec{E})$
- $[if\ E = E'\ then\ p\ else\ q](\vec{E}) = [q](\vec{E})$ if $[E](\vec{E}) \neq [E'](\vec{E})$
- $[case\ E\ of\ key\ do\ p\ else\ q](\vec{E}) = [p](\vec{E})$ if $[E](\vec{E}) \in \mathbf{Keys}$
- $[case\ E\ of\ key\ do\ p\ else\ q](\vec{E}) = [q](\vec{E})$ if $[E](\vec{E}) \notin \mathbf{Keys}$
- $[case\ E\ of\ x :: y\ do\ p\ else\ q](\vec{E}) = [p[h/x, t/y]](\vec{E})$ if $[E](\vec{E}) = h :: t$ with h atomic
- $[case\ E\ of\ x :: y\ do\ p\ else\ q](\vec{E}) = [q](\vec{E})$ if $[E](\vec{E}) = \varepsilon$

Model: Processes

Extend to streams per iteration:

Define $[p] : \mathbf{Stream}_{\tilde{I}} \to \mathcal{P}(\mathbf{Stream}_{\{c\}})$ by

$$[p](\vec{s}) \stackrel{\text{def}}{=} \{\vec{t} : \vec{t}_0 \in [p](\varepsilon, \ldots, \varepsilon) \wedge \forall n. \vec{t}_{n+1} \in [p](\vec{s}_n)\}.$$

Finally, $P = (I, O, L, (p_c)_{c \in \tilde{O}})$ gives $[\![P]\!] \stackrel{\text{def}}{=} \otimes_{c \in \tilde{O}} [p_c]$.

$[\![P]\!]$ causal: $n + 1^{st}$ output depends only on first n inputs.

Adversaries A may be non-causal: slightly different interpretation $[\![A]\!]_r$ (sometimes rushing adversaries, B. Pfitzmann).

Secrecy: Rely/guarantee

A obeys $C \subseteq \mathbf{Stream}_{O_P} \times \mathbf{Stream}_{I_P}$ if
for all $\vec{s} \in \mathbf{Stream}_{I_A}$ and $\vec{t} \in [\![A]\!](\vec{s})$, have $(\vec{s}\lfloor_O, \vec{t}\lfloor_I) \in C$.

Definition P preserves the secrecy of m assuming
$C \subseteq \mathbf{Stream}_{O_P} \times \mathbf{Stream}_{I_P}$ if exists no A obeying C such that
$[\![P]\!] \otimes [\![A]\!]_r$ may eventually output m (and $m \notin S_A \cup K_A$).

Example $p \stackrel{\text{def}}{=} \mathit{if}\ \ c = \mathbf{password}\ \mathit{then}\ \mathbf{secret}\ \mathit{else}\ \varepsilon$ preserves
secrecy of secret assuming $C = \{(\vec{t}, \vec{s}) : \forall n.\vec{s}_n \neq \mathbf{password}\}$.