Handling Encryption in an Analysis for Secure Information Flow

Peeter Laud

peeter_l@ut.ee

Tartu Ülikool

Cybernetica AS

Overview

- Some words about the overall approach.
- Definition of secure information flow.
  - also in computational sense.
- Ideas behind the analysis.
  - The domains that the analysis uses.
  - The abstraction.
  - Some examples.

Overall approach

- We have a program language containing encryption.
- We want to analyse programs for secure information flow.

Overall approach

- We have a program language containing encryption.
- We want to analyse programs for secure information flow.
- Use cryptographic definitions of secure encryption.
- Use the same domains in defining program semantics.
  - Bit-strings as values.
  - more... (the *security parameter*)
- Define secure information flow using cryptographic machinery.

Overall approach

- We have a program language containing encryption.
- We want to analyse programs for secure information flow.

- Use cryptographic definitions of secure encryption.
- Use the same domains in defining program semantics.
  - Bit-strings as values.
  - more. . . (the *security parameter*)
- Define secure information flow using cryptographic machinery.
- Devise the analysis.
  - Its proof of correctness has cryptographic nature.

Program language

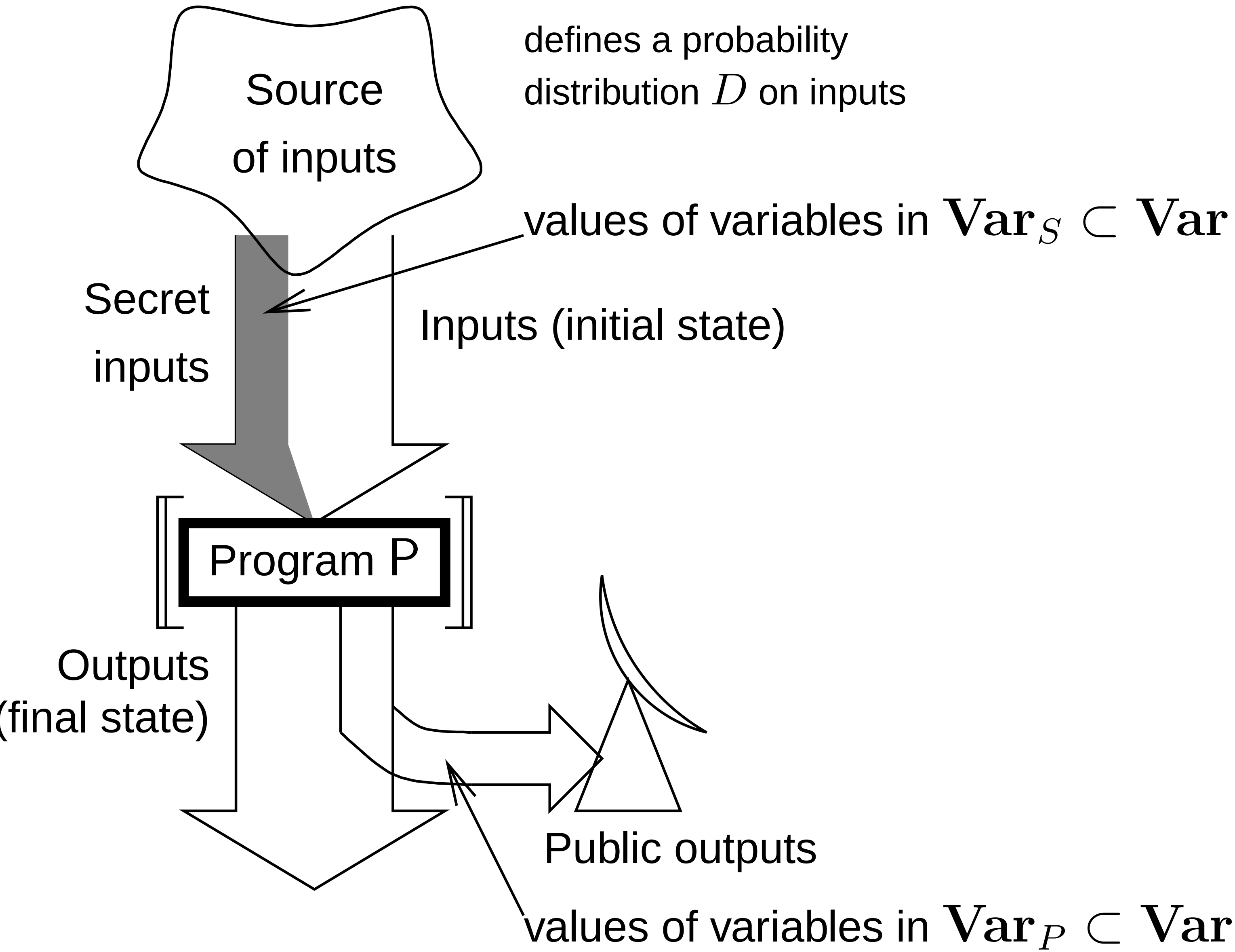
- The WHILE-language (simple imperative language).

$$
\begin{array}{lcl}
\mathsf{P} & ::= & x := o(x_1, \ldots, x_k) \\
 & | & \mathit{skip} \\
 & | & \mathsf{P}_1 ; \mathsf{P}_2 \\
 & | & \mathit{if}\ b\ \mathit{then}\ \mathsf{P}_1\ \mathit{else}\ \mathsf{P}_2 \\
 & | & \mathit{while}\ b\ \mathit{do}\ \mathsf{P}'
\end{array}
$$

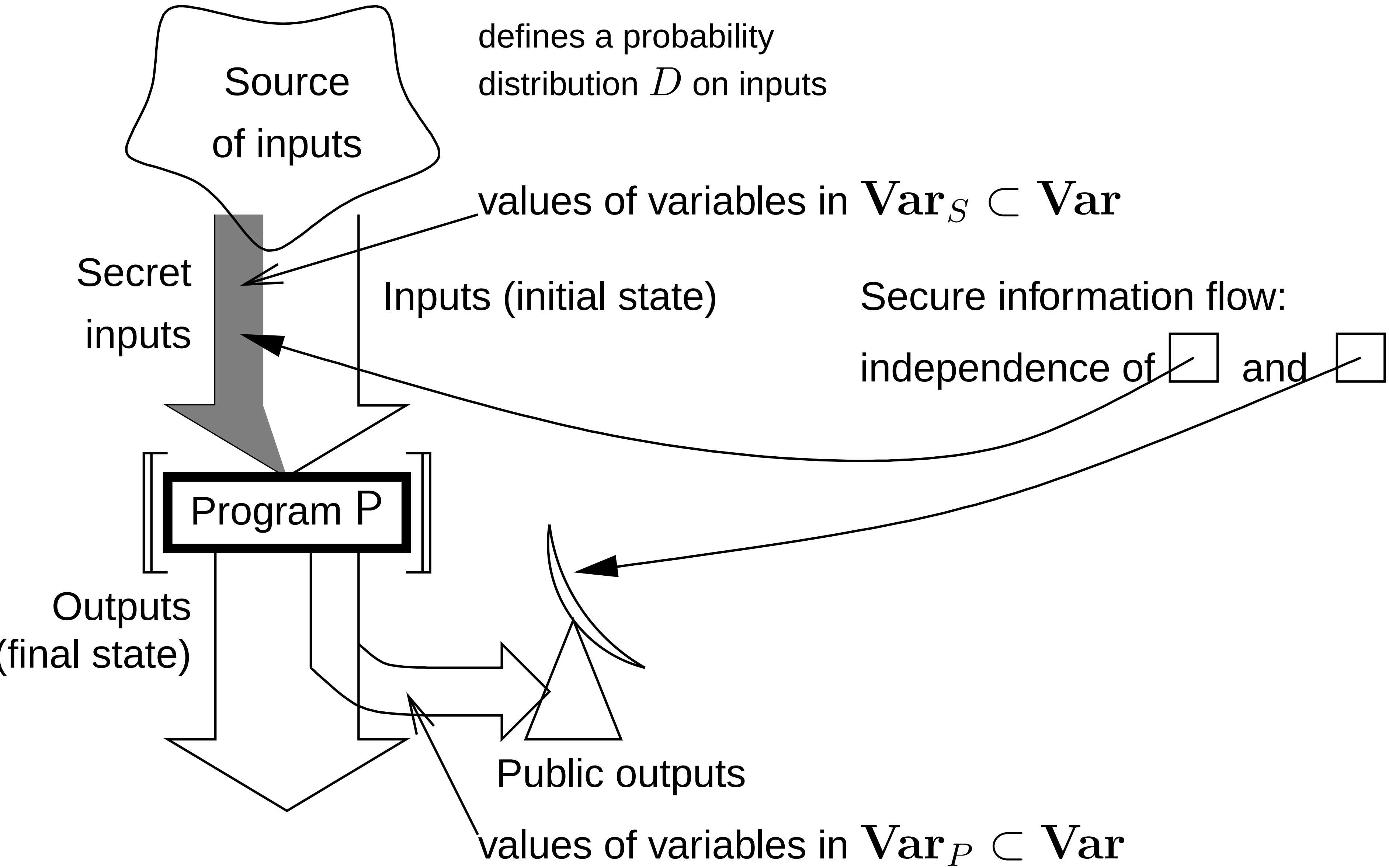
  $b, x, x_1, \ldots, x_k \in \mathbf{Var}$. $o \in \mathbf{Op}$. $\mathcal{E}nc, \mathcal{G}en \in \mathbf{Op}$.

- Denotational semantics, defined over program structure.
  - Maps initial state to final state.
  - Program state maps variables to their values.

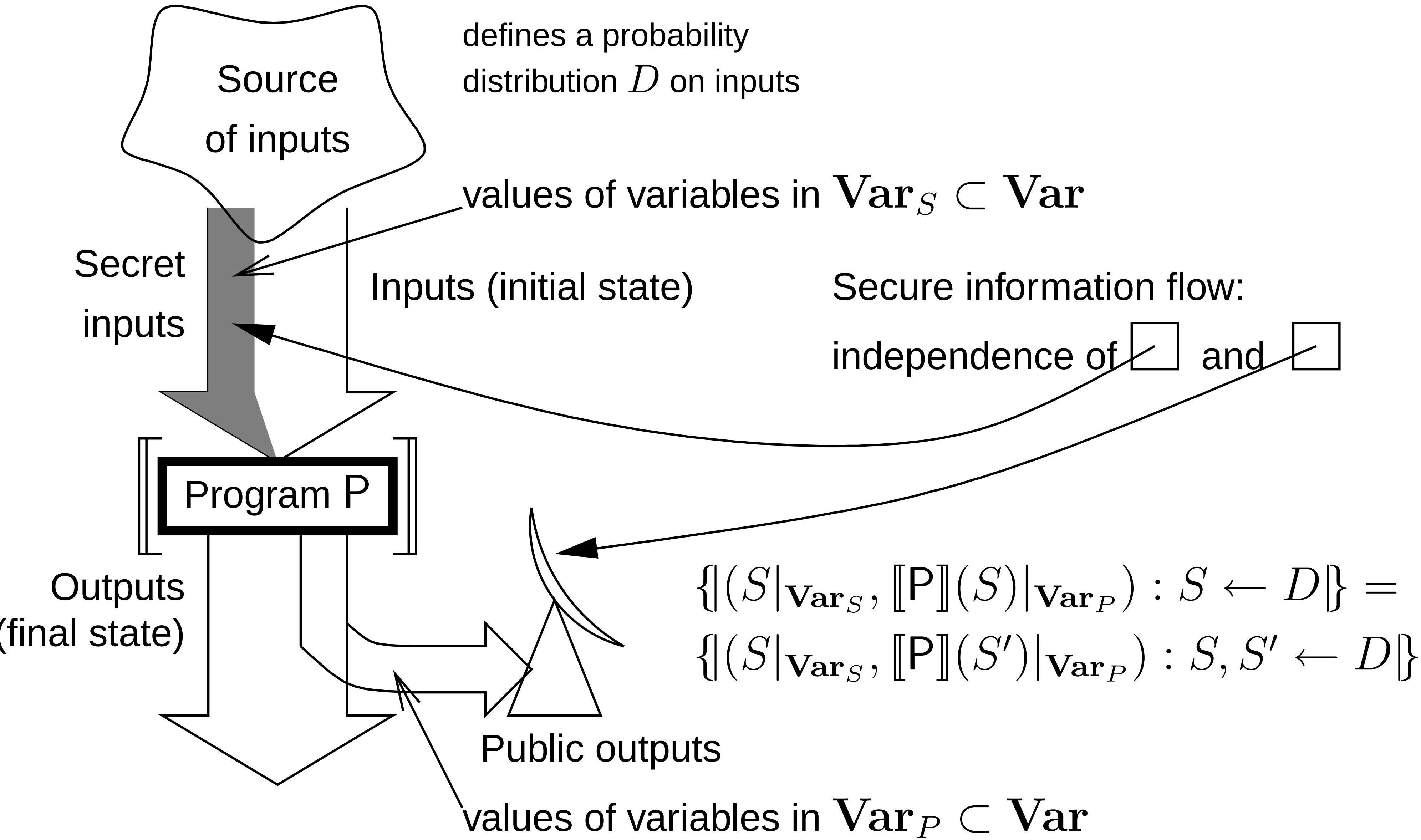
Security as independence



Security as independence



Security as independence



$$\{\!|(S|_{\mathbf{Var}_S}, [\![\mathsf{P}]\!](S)|_{\mathbf{Var}_P}) : S \leftarrow D|\!\} = \{\!|(S|_{\mathbf{Var}_S}, [\![\mathsf{P}]\!](S')|_{\mathbf{Var}_P}) : S, S' \leftarrow D|\!\}$$

Indistinguishability of distributions

- Let D^0, D^1 be two distributions over bit-strings.
- Let $\mathfrak{A}$ be a class of algorithms .
- Let $\mathcal{A} \in \mathfrak{A}$
- Consider the following experiment
  - Let $b \overset{R}{\in} \{0, 1\}$. Generate $x \leftarrow D^b$.
  - Run $\mathcal{A}(x)$. Let b^* be the output.

  Let $\mathbf{Adv}_{\mathcal{A}}^{D^0,D^1} = \Pr[b = b^*] - 1/2$.
- D^0 and D^1 are ε- *indistinguishable*, if $\mathbf{Adv}_{\mathcal{A}}^{D^0,D^1} \leq \varepsilon$ for all $\mathcal{A} \in \mathfrak{A}$.

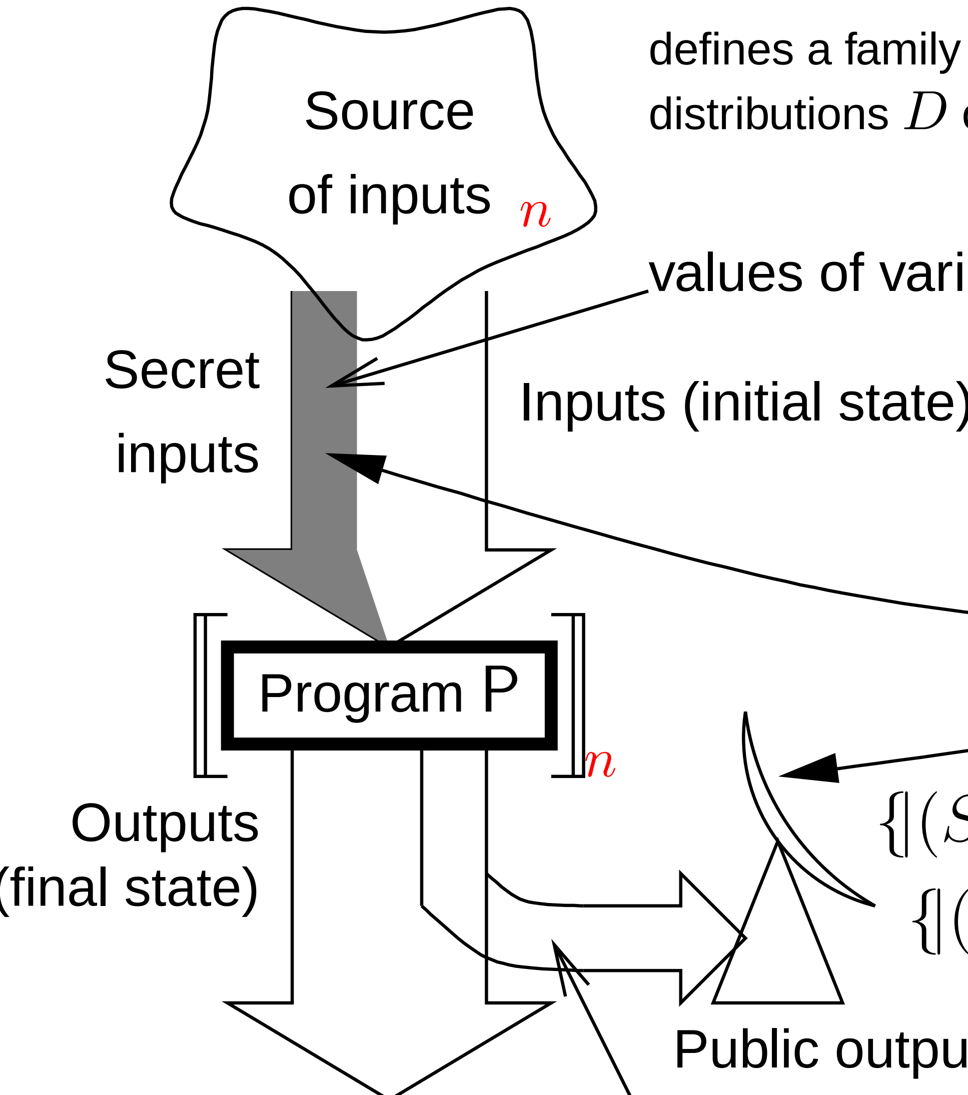
Indistinguishability of distributions

- Let $D^0 = \{D^0_n\}_{n\in\mathbb{N}}$, $D^1 = \{D^1_n\}_{n\in\mathbb{N}}$ be two families of distributions over bit-strings.
- Let $\mathfrak{A}$ be the class of algorithms running in poly-time.
- Let $\mathcal{A} \in \mathfrak{A}$
- Consider the following experiment
  - Let $b_n \overset{R}{\in} \{0,1\}$. Generate $x \leftarrow D^b_n$.
  - Run $\mathcal{A}(n,x)$. Let b^*_n be the output.

  Let $\mathbf{Adv}^{D^0,D^1}_{\mathcal{A}}(n) = \Pr[b_n = b^*_n] - 1/2$.
- D^0 and D^1 are *indistinguishable*, if $\mathbf{Adv}^{D^0,D^1}_{\mathcal{A}}$ is negligible for all $\mathcal{A} \in \mathfrak{A}$.

f is negligible $\overset{\text{def}}{\iff}$ $1/f$ is superpolynomial.

Definition of security



defines a family of probability distributions D on inputs

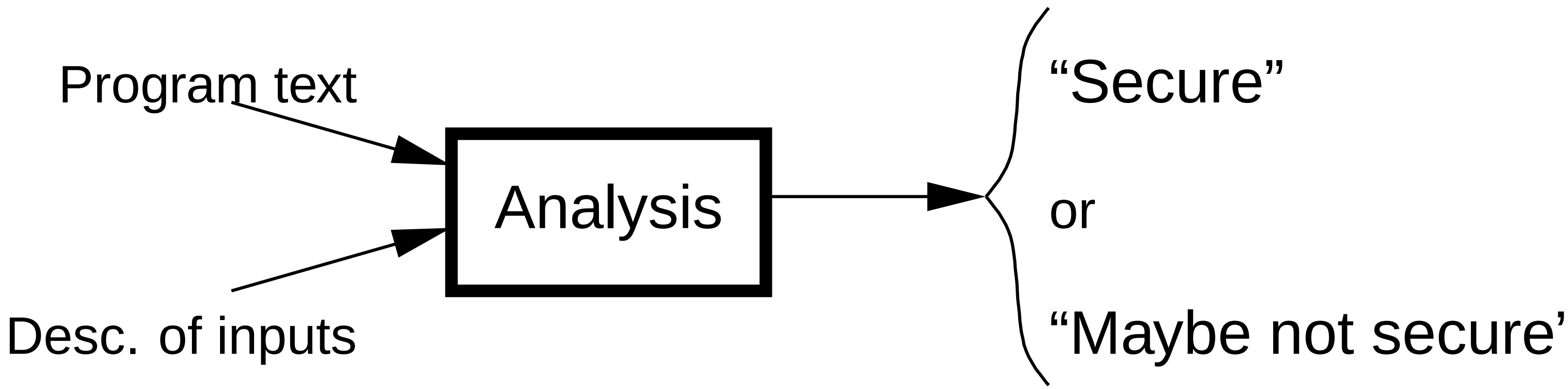
values of variables in $\mathbf{Var}_S \subset \mathbf{Var}$

Secure information flow:
independence of $\square$ and $\square$

$$\{\!|(S_n|_{\mathbf{Var}_S}, [\![\mathsf{P}]\!]_n(S_n)|_{\mathbf{Var}_P}) : S_n \leftarrow D_n|\!\} \approx \{\!|(S_n|_{\mathbf{Var}_S}, [\![\mathsf{P}]\!]_n(S'_n)|_{\mathbf{Var}_P}) : S_n, S'_n \leftarrow D_n|\!\}$$

values of variables in $\mathbf{Var}_P \subset \mathbf{Var}$

Program analysis' approach



- Having secure information flow is uncomputable in general.
- Description of inputs — whatever is known about D.
  - . . . and expressible in the domain of the analysis.

Domain of the analysis

- Analysis maps the description of the input distribution to the description of the output distribution.
- Description of $D = \{D_n\}_{n \in \mathbb{N}}$ is $(\mathcal{X}, \mathcal{K}) \in \mathcal{P}(\mathcal{P}(\mathbf{Var}) \times \mathcal{P}(\mathbf{Var})) \times \mathcal{P}(\mathbf{Var})$.
  - $(X, Y) \in \mathcal{X}$, if X and Y are independent in D.
  - $k \in \mathcal{K}$, if (the value of) k is distributed like a key.
- Assume the program does not change the variables in $\mathbf{Var}_S$.
- If $(\mathbf{Var}_S, \mathbf{Var}_P) \in \mathcal{X}_{\mathrm{output}}$, then the program has secure information flow.
- The analysis is defined inductively over the program structure.

Example: analysing assignments

Consider the program $x := o(x_1, \ldots, x_k)$.

If $(X \cup \{x_1, \ldots, x_k\}, Y) \in \mathcal{X}_{\mathrm{input}}$

then $(X \cup \{x_1, \ldots, x_k, x\}, Y) \in \mathcal{X}_{\mathrm{output}}$.

Analysing encryptions — problems

Let k be distributed like a key in D_{input}.

- Consider the program $l := k + 1$.

  Then $\{l\}$ is not independent of $\{k\}$ in D_{output}.
- Consider the program $x := \mathcal{E}nc(k, y)$.

  Then $\{x\}$ is not independent of $\{k\}$ in D_{output}.
  - To check whether x and k come from the same or from different samples of D_{output}, try to decrypt x with k.

These two cases should be distinguished as l is usable for decryption but x is not.

Encrypting black boxes

- Let $k \in \mathbf{Var}$. Let S be a program state.
- $S([k]_{\mathcal{E}})$ denotes a black box that encrypts with k. I.e.
  - $S([k]_{\mathcal{E}})$ has an input tape and an output tape;
  - When a bit-string w is written on the its tape,

$$[\![\mathcal{E}nc]\!](S(k), w)$$

  is invoked and the result written to the output tape.
- Indistinguishability can be defined for distributions over black boxes.
  - Independence can be defined, too.
- Security of $([\![\mathcal{G}en]\!], [\![\mathcal{E}nc]\!])$ is defined as the indistinguishability of certain black boxes.

Modified domain of the analysis

- Let $\widetilde{\mathbf{Var}} = \mathbf{Var} \uplus \{[x]_{\mathcal{E}} \,:\, x \in \mathbf{Var}\}$.
- Description of a distribution D is

$$(\mathcal{X}, \mathcal{K}) \in \mathcal{P}(\mathcal{P}(\widetilde{\mathbf{Var}}) \times \mathcal{P}(\widetilde{\mathbf{Var}})) \times \mathcal{P}(\mathbf{Var}) \quad .$$

  - $(X, Y) \in \mathcal{X}$ if X and Y are independent in D.
  - $k \in \mathcal{K}$, if the distribution of $[k]_{\mathcal{E}}$ according to D is indistinguishable from $[[\![\mathcal{G}en]\!]()]_{\mathcal{E}}$.

Analysing encryptions

Consider the program $x := \mathcal{E}nc(k, y)$.

If $(X, Y) \in \mathcal{X}_{\text{input}}$

and $k \in \mathcal{K}_{\text{input}}$

and $(\{[k]_{\mathcal{E}}\}, X \cup Y \cup \{y\}) \in \mathcal{X}_{\text{input}}$

then $(X \cup \{x\}, Y) \in \mathcal{X}_{\text{output}}$.

Generally $(\{[k]_{\mathcal{E}}\}, \{[k]_{\mathcal{E}}\}) \in \mathcal{X}_{\text{input}}$, hence $(\{x\}, \{[k]_{\mathcal{E}}\}) \in \mathcal{X}_{\text{output}}$.

If we have a program $l := k + 1$, then $(\{l\}, \{[k]_{\mathcal{E}}\}) \notin \mathcal{X}_{\text{output}}$.

For analysis of other program constructs see the article.

Concluding remarks

- Program analysis for computationally secure information flow.
- Based on abstracting
  - the families of probability distributions over program states
  - by pairs of sets of
    - variables and
    - encrypting black boxes

    that are independent of one another in it.
- No (non-trivial) constraints on program structure.
- Can be implemented.

`http://www.ut.ee/~peeter_l/research/csif`