Enforcing Dynamic Interference Policy

Frédéric Prost

LIG, Université de Grenoble
B. P. 53, F-38041 Grenoble, France
Frederic.Prost@imag.fr

Journées scientifiques SEmba 2011
21 Octobre 2011

Introduction

- Non-interférence comme socle mathématique pour la confidentialité.
- “Non-interference: who needs it ?” [Ryan et al. 01]
- Nombreux flots acceptables:
  - Cryptographie;
  - Vérification de mot de passe;
  - etc.

Introduction

- Non-interférence comme socle mathématique pour la confidentialité.
- “Non-interference: who needs it ?” [Ryan et al. 01]
- Nombreux flots acceptables:
  - Cryptographie;
  - Vérification de mot de passe;
  - etc.

$\Longrightarrow$ Politique d'interférence : la déclassification est permise dans certaines circonstances

- La politique est généralement statique.

Introduction

- De nombreux scénarios de la vie de tous les jours exhibent un comportement dynamique des niveaux de confidentialité.
  - Pay-per-view;
  - Enchères scellées;
  - etc.
- Challenge: adapter la non-interférence dans un cadre dynamique.

Introduction

- De nombreux scénarios de la vie de tous les jours exhibent un comportement dynamique des niveaux de confidentialité.
  - Pay-per-view;
  - Enchères scellées;
  - etc.
- Challenge: adapter la non-interférence dans un cadre dynamique.
- Nous proposons:
  1. Un "profil de sécurité" pour chaque opérateur : règles de réécritures sur les niveaux de confidentialité.
  2. Les règles de réécritures comportent des actions qui peuvent modifier la politique de confidentialité.
  3. Définition d'une "high/low bisimulation" par rapport à une politique de confidentialité.
  4. Vérification de la sécurité d'un programme par exécution abstraite.

Plan

Langage de programmation WHILE

- Langage de programmation minimaliste:

$$\begin{array}{lll} v & ::= & x \mid 0 \mid 1 \mid \mathsf{tt} \mid \mathsf{ff} \mid \ldots \\ t, b & ::= & v \mid f(x_1, \ldots, x_n) \\ P & ::= & x \leftarrow t \mid P; P \mid \\ & & \text{if } b \text{ then } P \text{ else } P \mid \text{while } b \text{ do } P \mid \text{skip} \end{array}$$

- On peut le voir comme un langage intermediaire:

$$x := f(345, g(x_1, x_2)) \equiv (x_0 \leftarrow 345; x_3 \leftarrow g(x_2, x_3); x \leftarrow f(x_3)$$

- Sémantique naturelle $\langle \mu, P \rangle \rightarrow_{\mathsf{os}} \langle \mu', P' \rangle$

Plan

Politique d'interférence

- Les variables du programmes sont étiquetées par des niveaux de confidentialité.
- Les niveaux de confidentialité sont élément d'un treillis de $\mathfrak{L}$.
- Les politiques d'interférences sont basées sur une description des comportement autorisés des opérateurs.

$\Longrightarrow$ Utilisation d'un système de réécriture sur les niveaux de confidentialité, ce qui permet de considérer les niveaux concrets au moment de l'évaluation du programme.

Politique d'interférence statique

- Pour chaque operateur f nous considérons f_{DIP}.
- $\Sigma_{DIP} = (\mathcal{V}_{DIP}, \mathcal{V} \cup \Omega_{DIP} \cup \mathfrak{L})$
- Politque d'encryptage, $\mathcal{D}$:

$$\begin{aligned}
&\text{encrypt}_{DIP}(\pi_{128}, \overline{x}) \rightarrow \pi_1 \\
&\text{encrypt}_{DIP}(\pi_{256}, \overline{x}) \rightarrow \bot \\
&\text{SPY}_{DIP} \rightarrow \bot \\
&\text{PIN}_{DIP} \rightarrow \top \\
&\ldots
\end{aligned}$$

- Dans le programme: $\text{SPY} \leftarrow \text{encrypt}(K, \text{PIN})$

  Les niveauw de confidentialité de $\text{encrypt}(K, \text{PIN})$ et SPY sont calculés en utilisant les règles de $\mathcal{D}$.

Aspects dynamiques

- Les niveaux de confidentialité assignés à une variable changent au cours de l'exécution du programme.
- Règles de réécriture avec actions : $l \to r; \mathsf{a}$

$$\mathsf{a} ::= x \mapsto \pi \mid \overline{x} \mapsto \pi \mid \overline{x} \mapsto \overline{y} \mid x \mapsto \overline{y} \mid \mathsf{a}; \mathsf{a}$$

- La politique d'interférence change avec l'évaluation du niveau de confidentialité des opérateurs :

$$\langle t[\sigma(l)], \mathcal{D} \rangle \rightsquigarrow \langle t[\sigma(r)], \mathcal{D}' \rangle$$

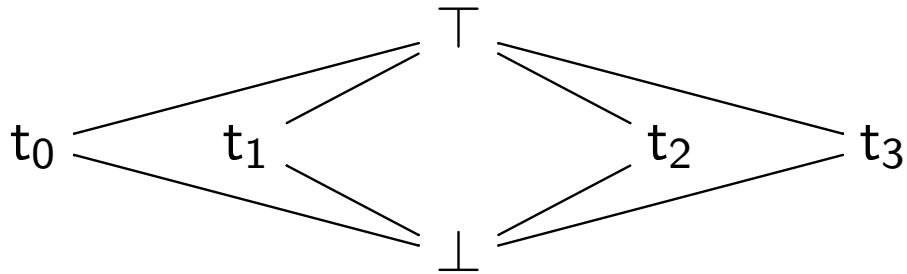
Three strikes, out

- Objectif: compte suspendu après 3 tentatives de connexion ratées.
- Dans le programme: ckpwd(g, PWD)
- Pour chaque opérateur f d'arité n, on considère f_{DIP} d'arité $2n$.

  $\Longrightarrow$ distinction entre le nom d'une variable dans le programme et son niveau de confidentialité.
- Le niveau de confidentialité de ckpwd(g, PWD) est calculé par l'évaluation de :

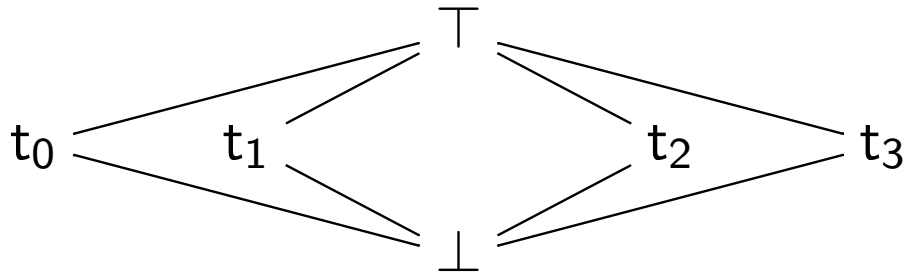
$$\mathsf{ckpwd}_{DIP}(\pi_g, g, \pi_{pwd}, \mathrm{PWD})$$

Three strikes, out



$\mathsf{ckpwd}(\bot, \overline{g}, \mathsf{t}_0, \overline{p}) \to \bot; \overline{p} \to \mathsf{t}_1$
$\mathsf{ckpwd}(\bot, \overline{g}, \mathsf{t}_1, \overline{p}) \to \bot; \overline{p} \to \mathsf{t}_2$
$\mathsf{ckpwd}(\bot, \overline{g}, \mathsf{t}_2, \overline{p}) \to \bot; \overline{p} \to \mathsf{t}_3$
$\mathsf{ckpwd}(\bot, \overline{g}, \mathsf{t}_3, \overline{p}) \to \top$
$\mathsf{ckok}(\overline{x}, \overline{y}) \to \bot; \overline{y} \to \mathsf{t}_0$
$g \to \bot$
$\mathrm{PIN}_1 \to \mathsf{t}_0$
$\mathrm{PIN}_2 \to \mathsf{t}_0$
$\ldots$

Three strikes, out



$\mathsf{ckpwd}(\bot, \overline{g}, \mathsf{t}_0, \overline{p}) \to \bot; \overline{p} \to \mathsf{t}_1$
$\mathsf{ckpwd}(\bot, \overline{g}, \mathsf{t}_1, \overline{p}) \to \bot; \overline{p} \to \mathsf{t}_2$
$\mathsf{ckpwd}(\bot, \overline{g}, \mathsf{t}_2, \overline{p}) \to \bot; \overline{p} \to \mathsf{t}_3$
$\mathsf{ckpwd}(\bot, \overline{g}, \mathsf{t}_3, \overline{p}) \to \top$
$\mathsf{ckok}(\overline{x}, \overline{y}) \to \bot; \overline{y} \to \mathsf{t}_0$
$g \to \bot$
$\mathrm{PIN}_1 \to \mathsf{t}_0$
$\mathrm{PIN}_2 \to \mathsf{t}_0$
...

Dans le programme:

if $\mathsf{ckpwd}(g, \mathrm{PIN}_1)$ then blah else next_try

Niveau de confidentialité d'un terme par rapport à $\mathcal{D}$

- Pour le calcul du niveau de confidentialité de $f(x, y)$ on considère

$$t = f_{DIP}(nf^{\mathcal{D}}(x), x, nf^{\mathcal{D}}(y), y)$$

- L'évaluation de ce terme dans $\mathcal{D}$ donne le niveau de confidentialité et une nouvelle politique.

$$\langle t, \mathcal{D} \rangle \rightsquigarrow^* \langle \pi^{\mathcal{D}}(t), \overline{\mathcal{D}}^t \rangle$$

Plan

Sécurité d'un programme

- Traditionallement: un programme est sûr si chaque modification d'une valeur au dessus de π ne peut être observée en dessous de π :

$$\begin{array}{l} \langle \mu_1, P \rangle \rightarrow^*_{\mathsf{os}} \mu'_1 \\ \langle \mu_2, P \rangle \rightarrow^*_{\mathsf{os}} \mu'_2 \\ \mu'_1 \equiv_\pi \mu'_2 \end{array}$$

- Que faire avec la politique:

$$\mathsf{encrypt}(\pi_{1024}, \top) \rightarrow \bot$$

rendant possible le programme :

$$\mathrm{SPY} \leftarrow \mathsf{encrypt}(\mathit{key}_{1024}, \mathrm{PIN})$$

Sécurité d'un programme

- Traditionallement: un programme est sûr si chaque modification d'une valeur au dessus de π ne peut être observée en dessous de π :

$$\begin{array}{l}\langle \mu_1, P\rangle \rightarrow^*_{\mathsf{os}} \mu_1' \\ \langle \mu_2, P\rangle \rightarrow^*_{\mathsf{os}} \mu_2' \\ \mu_1' \equiv_\pi \mu_2'\end{array}$$

- Que faire avec la politique:

$$\mathsf{encrypt}(\pi_{1024}, \top) \rightarrow \bot$$

rendant possible le programme :

$$\mathrm{SPY} \leftarrow \mathsf{encrypt}(\mathit{key}_{1024}, \mathrm{PIN})$$

$\Longrightarrow$ Sémantique opérationnelle alternative : les fuites d'informations déclarées sont traitées de manière spécifique.

- Notion de sémantique opérationnelle de déclassification.

$$\langle \mu_1, P\rangle \stackrel{\mu_d}{\Longrightarrow} \langle \mu_1', P'\rangle$$

Termes déclassifiants

- Un terme est déclassifyiant si son niveau de confidentialité est inférieur à celui d'un de ses arguments.
- Ces termes seront soumis à des règles d'évaluation spécifique par rapport à la sémantique opérationnelle de déclassification.

Definition (Termes déclassifiants)

$t = f(x_1, \ldots, x_n)$ est déclassifiant par rapport $\mathcal{D}$, ce qu'on écrit $\mathcal{D} \vdash f(x_1, \ldots, x_n) \downarrow$ si:

$$\pi^{\mathcal{D}}(t) \sqsubseteq (\bigsqcup_{i=1}^{n} \pi^{\mathcal{D}}(t_i))$$

High/low bisimulation et politique dynamique

Definition (Bisimulation)

Une π-bisimulation est une relation symétrique $\mathcal{R}$:

If $\langle P_1, \mathcal{D}_1 \rangle \mathcal{R} \langle P_2, \mathcal{D}_2 \rangle$ and

$\langle \mu_1, P_1, \mathcal{D}_1 \rangle \stackrel{\mu_1}{\Longrightarrow} \langle \mu_1', P_1', \mathcal{D}_1' \rangle$

and $\mu_1 \simeq_{\pi}^{\mathcal{D}_1 \sqcup \mathcal{D}_2} \mu_2$

High/low bisimulation et politique dynamique

Definition (Bisimulation)

Une π-bisimulation est une relation symétrique $\mathcal{R}$:

$$\begin{array}{l} \text{If } \langle P_1, \mathcal{D}_1 \rangle \mathcal{R} \langle P_2, \mathcal{D}_2 \rangle \text{ and} \\ \\ \langle \mu_1, P_1, \mathcal{D}_1 \rangle \xRightarrow{\mu_1} \langle \mu_1', P_1', \mathcal{D}_1' \rangle \\ \\ \text{and } \mu_1 \simeq_\pi^{\mathcal{D}_1 \sqcup \mathcal{D}_2} \mu_2 \end{array} \implies \begin{cases} \exists P_2', \mathcal{D}_2' \text{ and } \mu_2' \text{ s.t.} \\ \\ \langle \mu_2, P_2, \mathcal{D}_2 \rangle \xRightarrow{\mu_1}{}^{*} \langle \mu_2', P_2', \mathcal{D}_2' \rangle \\ \\ \text{and } \mu_1' \simeq_\pi^{\mathcal{D}_1' \sqcup \mathcal{D}_2'} \mu_2' \\ \\ \text{and } \langle P_1', \mathcal{D}_1' \rangle \mathcal{R} \langle P_2', \mathcal{D}_2' \rangle \end{cases}$$

Sécurité des programmes par rapport à une politique dynamique

Definition (Programme sûr)

Un programme P est sûre vis-à-vis de $\mathcal{D}$, ce qu'on écrit $\mathcal{D} \models P$, si pour chaque niveau de confientialité π $\langle P, \mathcal{D} \rangle \simeq^{\pi} \langle P, \mathcal{D} \rangle$.

- Cette définition est une extension conservative de résultats précédents:
  - Sans actions $\Longrightarrow$ [EchahedProst05].
  - Règles de réécritures limités à la bsup des arguments $\Longrightarrow$ non-interférence standard.

Plan

Principe d'exécution abstraite(1)

Principe d'exécution abstraite(1)

- Idée: exécuter le programme sur $\mathfrak{L}$.
- Une mémoire abstraite associe chaque variable du programme à son niveau de confidentialité.

Principe d'exécution abstraite(1)

- Idée: exécuter le programme sur $\mathfrak{L}$.
- Une mémoire abstraite associe chaque variable du programme à son niveau de confidentialité.
- Il faut enregitrer le plus haut niveau de confidentialité rencontré dans les gardes :

$$\text{if } \textsc{pin} = 0 \text{ then while tt do skip else skip; } \textsc{spy} \leftarrow 0$$

- Vérification des affectations vis-à-vis de $\mathcal{D}$ et :

$$x \leftarrow f(\ldots) \text{ implique } \pi^{\mathcal{D}}(x) \subseteq (\pi^{\mathcal{D}}(f(\ldots)) \sqcup \pi_g)$$

Principe d'exécution abstraite(2)

- L'évaluation des termes modifie la politique.
- Problème: il n'est pas possible de mélanger les politiques.

Principe d'exécution abstraite(2)

- L'évaluation des termes modifie la politique.
- Problème: il n'est pas possible de mélanger les politiques.

  $\Longrightarrow$ Création d'une liste de politiques pour chaque chemin d'exécution.
- Problème de point fixe pour le while.

Principe d'exécution abstraite(2)

- L'évaluation des termes modifie la politique.
- Problème: il n'est pas possible de mélanger les politiques.

  $\Longrightarrow$ Création d'une liste de politiques pour chaque chemin d'exécution.
- Problème de point fixe pour le while.

  $\Longrightarrow$ Nombre fini de listes de politiques.

Résultats sur l'exécution abstraite

Theorem

$$\exists \mathcal{L}.(\langle \{\langle \mathcal{D}, \bot \rangle\}, P \rangle \hookrightarrow^* \mathcal{L}) \implies \mathcal{D} \models P$$

Résultats sur l'exécution abstraite

Theorem

$$\exists \mathcal{L}.(\langle\{\langle \mathcal{D}, \bot\rangle\}, P\rangle \hookrightarrow^* \mathcal{L}) \implies \mathcal{D} \models P$$

- Implication inverse ne tient pas (et ne peut pas) :

$$\text{if } \textsc{pin} = 0 \text{ then } \textsc{spy} \leftarrow 1 \text{ else } \textsc{spy} \leftarrow 1$$

Plan

Conclusion

- Nous avons introduit de la dynamicité dans les politiques de confidentialité.
  - Définition des niveaux de confidentialité des termes.
  - Définition d'un programme sûr par rapport à une sécurité.
  - Analyse de programme correcte (mais non complète).

  $\Longrightarrow$ Ce cadre permet de prendre en compte beaucoup de scénarios réalistes que les approches traditionnelles ne peuvent traiter.

Conclusion

- Nous avons introduit de la dynamicité dans les politiques de confidentialité.
  - Définition des niveaux de confidentialité des termes.
  - Définition d'un programme sûr par rapport à une sécurité.
  - Analyse de programme correcte (mais non complète).

  $\Longrightarrow$ Ce cadre permet de prendre en compte beaucoup de scénarios réalistes que les approches traditionnelles ne peuvent traiter.
- Etude théorique: que se passe-t-il sur des programmes réalistes.
- Aspects dynamiques limités aux évolutiond des niveaux de confidentialité.
- Extension en présence de concurrence ?
- Parallélisation, program slicing ?

Merci de votre attention

Questions ?