Another look inside Multiple Facets

M. Ngo [1] N. Bielova [1] C. Flanagan [2] T. Rezk [1]
A. Russo [3] T. Schmitz [2]

[1]INRIA, France
[2]UCSC, USA
[3]Chalmers, Sweden

May 10, 2017

Secure Multi-Execution and Multiple Facets

Secure Multi-Execution (SME) [Devriese and Piessens 2010]	**Multiple Facets (MF)** [Austin and Flanagan 2012]
Black Box: does not look into programs.	Not Black Box
Constructed for a finite lattice. For a lattice with n levels, has n executions.	Constructed for a finite set of principals. For k principals, has *less than* 2^k executions (while SME has 2^k executions). MF is more resource friendly than SME).
Offers termination insensitive non-interference and a version of termination sensitive non-interference.	Offers only termination insensitive non-interference.

Research Question 1

Can we make Multiple Facets (MF) more black box and still be resource friendly?

- The advantage of MF over SME is based on looking inside programs.
- It is not possible to make MF more black box without loosing its advantage.
- MF can be optimized further. Propose *Optimized Generalized Multiple Facets (OGMF)*.

Research Question 2

Compare SME and MF in the terms of the number of executions.

- The comparison in [Austin and Flanagan 2012] is incomplete.
  - Lattice $\langle \mathcal{L}, \sqsubseteq \rangle$ where $\mathcal{L} = \{L, M, H\}$ and $L \sqsubseteq M \sqsubseteq H$.
  - SME on the lattice: 3 executions.
  - To encode the lattice with the set of principals: need two principals k_1 and k_2.
  - The lattice with these two principals is $\langle \mathcal{L}^\bullet, \subseteq \rangle$, where $\mathcal{L}^\bullet = \{\emptyset, \{k_1\}, \{k_2\}, \{k_1, k_2\}\}$.
    M can be encoded as $\{k_1\}$ or $\{k_2\}$.
  - MF for $\{L, M, H\}$: may have at most 4 executions.
- Propose *Generalized Multiple Facets (GMF)*, a version of MF for arbitrary finite lattices.

Research Question 3

Can Multiple Facets (MF) prevent leakage on termination channel?

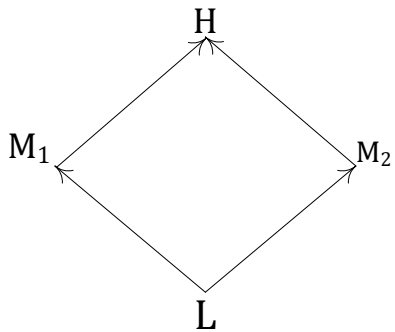
- Leakage on termination channel: confidential data influence termination of executions.
- MF does not prevent leakage on termination channel.
- Propose *Termination Sensitive Multiple Facets (TSMF)*.

Outline

Notation

$$P ::= \textbf{skip} \mid x := e \mid P; P \mid \qquad \text{program}$$
$$\mid \textbf{if } e \textbf{ then } P \textbf{ else } P \mid \textbf{while } e \textbf{ do } P$$
$$e ::= v \mid x \mid e \oplus e \qquad \text{expression}$$

- A memory μ is a mapping from variables to simple values.
- The semantics is standard.
- The diamond lattice with four security levels: L, M_1, M_2 and H, where $L \sqsubseteq M_1 \sqsubseteq H$ and $L \sqsubseteq M_2 \sqsubseteq H$.



Faceted Value

A *faceted value* is of the form $\langle l \,?\, V_1 : V_2 \rangle$, where l is a security level and V_i is a faceted value or a simple value.

- V_1 is the *private facet* of V, and is visible at l', where $l \sqsubseteq l'$.
- V_2 is the *public facet* of V, and is visible at l', where $l \not\sqsubseteq l'$.

The *projection of a (faceted or simple) value V at level l* is $l(V)$. $l(V)$ is the visible (simple) value of V at l.

$$
\begin{aligned}
l(v) &= v \\
l(\langle l' \,?\, V_1 : V_2 \rangle) &= \begin{cases} l(V_1) & \text{if } l' \sqsubseteq l, \\ l(V_2) & \text{if } l' \not\sqsubseteq l. \end{cases}
\end{aligned}
$$

A *faceted memory* $\hat{\mu}$ is a mapping from variables to faceted values or simple values.

Example. Suppose that $V = \langle M_1 \,?\, \langle H \,?\, 1 : 2 \rangle : 3 \rangle$.

- The private facet is $\langle H \,?\, 1 : 2 \rangle$, the public facet is 3.
- $H(V) = H(\langle H \,?\, 1 : 2 \rangle) = H(1) = 1$. $M_1(V) = 2$. $L(V) = M_2(V) = 3$.
- $\hat{\mu}(x) = \langle M_1 \,?\, \langle H \,?\, 1 : 2 \rangle : 3 \rangle$, and $\hat{\mu}(y) = 0$.

Expression Evaluation

The evaluation of an expression e with a faceted memory $\hat{\mu}$ at level l is denoted by $\hat{\mu}^l(e)$.

- When e is a simple value v, $\hat{\mu}^l(v) = v$,
- When e is a variable x, $\hat{\mu}^l(x) = l(\hat{\mu}(x))$,
- When e is of the form $e_1 \oplus e_2$, only values visible at l are used in the evaluation.

Example. Suppose that $\hat{\mu}(x) = V_1 = \langle M_1 \,?\, 10 : 0 \rangle$ and $\hat{\mu}(y) = V_2 = \langle M_2 \,?\, 5 : 0 \rangle$.

- $\hat{\mu}^H(x) = 10$ since $H(\hat{\mu}(x)) = H(\langle M_1 \,?\, 10 : 0 \rangle) = 10$.
- $\hat{\mu}^H(y) = 5$ since $H(\hat{\mu}(y)) = H(\langle M_2 \,?\, 5 : 0 \rangle) = 5$.
- $\hat{\mu}^H(x + y) = 10 + 5 = 15$ since $H(\hat{\mu}(x)) = 10$ and $H(\hat{\mu}(y)) = 5$.

Expression Evaluation - Cont.

Generalize $\hat{\mu}^l(e)$ to $\hat{\mu}^{pc}(e)$, where *pc* is a set of security levels s.t. $\hat{\mu}^{pc}(e) = V$ iff for any l in *pc*, $\hat{\mu}^l(e) = l(V)$.

Example. Suppose that $\hat{\mu}(x) = V_1 = \langle M_1 \,?\, 10 : 0 \rangle$, $\hat{\mu}(y) = V_2 = \langle M_2 \,?\, 5 : 0 \rangle$, and $pc = \{M_1, H\}$.

- $\hat{\mu}^{pc}(x) = V_3 = 10$
  - verify: $H(V_1) = H(V_3) = 10$, and $M_1(V_1) = M_1(V_3) = 10$
- $\hat{\mu}^{pc}(y) = \langle M_2 \,?\, 5 : 0 \rangle$
- $\hat{\mu}^{pc}(x + y) = \langle M_2 \,?\, 15 : 10 \rangle$

Program Evaluation

- The terminating evaluation of a program P with a faceted memory $\hat{\mu}$ and a set of security levels pc is denoted by $(P, \hat{\mu}) \downarrow_G^{pc} \hat{\mu}'$.
- The semantics rule for assignment command is based on $\hat{\mu}^{pc}$.

$$\textsc{GAssign}\ \frac{\hat{\mu}^{pc}(e) = V}{(x := e, \hat{\mu}) \downarrow_G^{pc} \hat{\mu}[x \mapsto V]}$$

Program Evaluation - If

The evaluation of **if** e **then** P_{true} **else** P_{false} with $\hat{\mu}$ and pc.

- $\hat{\mu}^{pc}(e) = v$: evaluate only P_v.
- $\hat{\mu}^{pc}(e) = \langle l\,?\,V_1 : V_2 \rangle$
  - There exist $l_1 \in pc$ s.t. $l \sqsubseteq l_1$, there exists $l_2 \in pc$ s.t $l \not\sqsubseteq l_2$: split the evaluation to two - one with V_1, one with V_2, and then combine the results of the two evaluations.
  - $\forall l' \in pc : l \sqsubseteq l'$: evaluate **if** V_1 **then** P_{true} **else** P_{false}.
  - $\forall l' \in pc : l \not\sqsubseteq l'$: evaluate **if** V_2 **then** P_{true} **else** P_{false}

$$\textsc{GIF-C}\ \frac{\hat{\mu}^{pc}(e) = v \qquad (P_v, \hat{\mu}) \downarrow_G^{pc} \hat{\mu}'}{(\textbf{if } e \textbf{ then } P_{true} \textbf{ else } P_{false}, \hat{\mu}) \downarrow_G^{pc} \hat{\mu}'}$$

$$\textsc{GIF-S}\ \frac{\begin{array}{c}\hat{\mu}^{pc}(e) = \langle l\,?\,V_1 : V_2 \rangle \qquad pc_1 = \{l' \in pc | l \sqsubseteq l'\} \neq \emptyset \\ pc_2 = pc \setminus pc_1 \neq \emptyset \qquad (\textbf{if } V_1 \textbf{ then } P_{true} \textbf{ else } P_{false}, \hat{\mu}) \downarrow_G^{pc_1} \hat{\mu}'_1 \\ (\textbf{if } V_2 \textbf{ then } P_{true} \textbf{ else } P_{false}, \hat{\mu}) \downarrow_G^{pc_2} \hat{\mu}'_2\end{array}}{(\textbf{if } e \textbf{ then } P_{true} \textbf{ else } P_{false}, \hat{\mu}) \downarrow_G^{pc} \hat{\mu}'_1 \otimes^{pc_1} \hat{\mu}'_2}$$

Note: to simplify the presentation, faceted values are considered as expressions.

Program Evaluation - If (Cont.)

Example. Suppose $\hat{\mu}(x) = \langle M_1 \,?\, \mathit{true} : \mathit{false} \rangle$, and $pc = \{L, M_1, M_2, H\}$. Consider the evaluation of

$$\textbf{if } x \textbf{ then } z := 5 \textbf{ else } z := 10$$

- $\hat{\mu}^{pc}(x) = \langle M_1 \,?\, \mathit{true} : \mathit{false} \rangle$
- $pc_1 = \{l \in pc | M_1 \sqsubseteq l\} = \{M_1, H\}$, $pc_2 = pc \setminus pc_1 = \{L, M_2\}$
- The evaluation is split to two.
  - **if** *true* **then** $z := 5$ **else** $z := 10$ is evaluated with pc_1. Its result is $\hat{\mu}'_1$, where $\hat{\mu}'_1(z) = 5$.
  - **if** *false* **then** $z := 5$ **else** $z := 10$ is evaluated with pc_2. Its result is $\hat{\mu}'_2$, where $\hat{\mu}'_2(z) = 10$.
  - $\hat{\mu}'_1$ and $\hat{\mu}'_2$ are combined to $\hat{\mu}'$, where $\hat{\mu}'(z) = \langle M_1 \,?\, 5 : 10 \rangle$.

Generalized Multiple Facets

Let Γ be a *security environment*, a mapping from variables to security levels in a lattice $\langle \mathcal{L}, \sqsubseteq \rangle$. Let $\bot$ be the lowest level in the lattice, *df* be a function that maps variables to default values. To apply GMF on a program P and a memory μ:

- Convert μ to a faceted memory $\hat{\mu}$.

$$\hat{\mu}(x) = \begin{cases} \mu(x) & \text{if } \Gamma(x) = \bot, \\ \langle \Gamma(x) \,?\, \mu(x) : df(x) \rangle & \text{otherwise.} \end{cases}$$

- Evaluate P with $\hat{\mu}$ at $pc = \mathcal{L}$. Suppose that $(P, \hat{\mu}) \downarrow_G^{\mathcal{L}} \hat{\mu}'$.
- Convert $\hat{\mu}'$ to a memory based on levels of variables.

$$\text{GMF} \ \frac{(P, \hat{\mu}) \downarrow_G^{\mathcal{L}} \hat{\mu}'}{\Gamma \vdash (P, \mu) \Downarrow_G \mu'}$$

Theorem

GMF satisfies termination insensitive non-interference.

Outline

GMF - Optimized further

Suppose $\hat{\mu}(x) = \langle M_1 \,?\, \langle H \,?\, \mathit{true} : \mathit{false} \rangle : \mathit{false} \rangle$, and $pc = \{L, M_1, M_2, H\}$. Consider the evaluation of

$$\textbf{if } x \textbf{ then } z := 5 \textbf{ else } z := 10.$$

- **if** $\langle H \,?\, \mathit{true} : \mathit{false} \rangle$ **then** $z := 5$ **else** $z := 10$ with $pc_1 = \{H, M_1\}$: split.
  - $z := 5$ is evaluated with $pc_{11} = \{H\}$. The result is $\hat{\mu}'_{11}$.
  - $z := 10$ is evaluated with $pc_{12} = \{M_1\}$. The result is $\hat{\mu}'_{12}$.
- **if** *false* **then** $z := 5$ **else** $z := 10$ with $pc_2 = \{L, M_2\}$. Hence, $z := 10$ is evaluated with pc_2. The result is $\hat{\mu}'_2$.
- Three sub-evaluations: one for $z := 5$, two for $z := 10$.
- Two times of combining faceted memories: $\hat{\mu}'_{11}$ is combined with $\hat{\mu}'_{12}$, and the result is combined with $\hat{\mu}'_2$

By merging the evaluations with pc_{11} and pc_2 (the evaluations where $z := 10$), we reduce the number of evaluations to two and the number of times of combining faceted memories to one.

Semantics

Evaluation of **if** e **then** P_{true} **else** P_{false} with $\hat{\mu}$ and pc:
$\hat{\mu}^{pc}(e) = V$

- if for all $l \in pc$, $l(V) = true$: evaluate only P_{true}.
- if for all $l \in pc$, $l(V) \neq true$: evaluate only P_{false}.
- otherwise, evaluate P_{true} and P_{false} and combine the results of these two evaluations.

$$\text{OIF-T}\ \frac{\hat{\mu}^{pc}(e) = V \qquad pc_1 = \{l \in pc | l(V) = true\} \qquad pc_1 = pc \qquad (P_{true}, \hat{\mu}) \downarrow_O^{pc} \hat{\mu}'}{(\textbf{if } e \textbf{ then } P_{true} \textbf{ else } P_{false}, \hat{\mu}) \downarrow_O^{pc} \hat{\mu}'}$$

$$\text{OIF-S}\ \frac{\begin{array}{c}\hat{\mu}^{pc}(e) = V \qquad pc_1 = \{l \in pc | l(V) = true\} \qquad pc_2 = pc \setminus pc_1 \\ pc_1 \neq \emptyset \qquad pc_2 \neq \emptyset \qquad (P_{true}, \hat{\mu}) \downarrow_O^{pc_1} \hat{\mu}'_1 \qquad (P_{false}, \hat{\mu}) \downarrow_O^{pc_2} \hat{\mu}'_2\end{array}}{(\textbf{if } e \textbf{ then } P_{true} \textbf{ else } P_{false}, \hat{\mu}) \downarrow_O^{pc} \hat{\mu}'_1 \oplus^{pc_1, pc_2} \hat{\mu}'_2}$$

Properties

Theorem

OGMF and GMF are equivalent.

Corollary

OGMF satisfies termination insensitive non-interference.

Outline

GMF, OGMF, and Termination Channel

GMF and OGMF do not prevent leakage on termination channel.

Example. Suppose that $\mathcal{L} = \{L, H\}$, $\Gamma(x) = H$. Consider the evaluation of the below program with GMF and OGMF.

if x **then** (**while** *true* **do skip**) **else skip**

- With $\hat{\mu}_1 = \hat{\mu}[x \mapsto \langle H \,?\, \mathit{true} : \mathit{false} \rangle]$, the evaluation diverges since the while loop is evaluated.
- With $\hat{\mu}_2 = \hat{\mu}[x \mapsto \langle H \,?\, \mathit{false} : \mathit{false} \rangle]$, the evaluation converges.
- Based on observations on those two evaluations, an attacker at L can gain insight about the high facet of x.

Termination Sensitive Multiple Facets

- Combine OGMF and SME
- When an if command is evaluated, TSMF performs a bounded evaluation of the command by using OGMF.
  - If the bounded evaluation terminates within a given time bound, the result of the bounded evaluation is used.
  - Otherwise, the command is evaluated with SME.
- Similar to SME, TSMF offers a version of termination sensitive non-interference.

$$\textsf{TIF1}\ \frac{(\textbf{if}\ e\ \textbf{then}\ P_1\ \textbf{else}\ P_2, \hat{\mu}) \Downarrow^{D}_{Bnd} \hat{\mu}'}{(\textbf{if}\ e\ \textbf{then}\ P_1\ \textbf{else}\ P_2, \hat{\mu}) \rightarrow_T (\textbf{skip}, \hat{\mu}')}$$

$$\textsf{TIF2}\ \frac{P = \textbf{if}\ e\ \textbf{then}\ P_1\ \textbf{else}\ P_2 \qquad (P, \hat{\mu}) \Downarrow^{D}_{Bnd} \bot}{(\textbf{if}\ e\ \textbf{then}\ P_1\ \textbf{else}\ P_2, \hat{\mu}) \rightarrow_T \text{toSME}(P, \hat{\mu})}$$

Summary

- We extended MF to work with arbitrary finite lattices (GMF) based on the observation that lattices with principals are not always the most convenient ones to use since they may create more executions than needed.
- Knowing all the points in the lattice allows for further optimization: creating executions could be done on a value-based basis (OGMF) rather than on security levels (as in GMF).
- We proposed a hybrid approach which present a balance between resource usage and security guarantees: it behaves as OGMF as long as it can and switches to SME when termination leaks might occur (TSMF).

Thank you for your attention.