Type-based information declassification

(in-development)

Presented by Alex Coleman

TYPDSEC project: “Type-based information declassification and its secure compilation”

TYPDSEC project in Collaboration with:

Alex Coleman (University of Kent),
Vineet Rajani (University of Kent),
Hrutvik Kanabar and
Magnus Myreen (Chalmers University of Technology)

University of Kent

Main related work

- DCC (A Core Calculus of Dependency) [1]
    - Introduces a modal type for classification
- CG (Types for Information Flow Control: Labelling Granularity and Semantic Models) [2] –
    - Uses semantic models and logical relations to prove soundness Hyper-property - (Non-interference)

[1] Abadi et al. 1999. A core calculus of dependency. In Proceedings of the 26th ACM SIGPLAN-SIGACT symposium on Principles of programming languages (POPL '99)

[2] Rajani et al, "Types for Information Flow Control: Labeling Granularity and Semantic Models," 2018 IEEE 31st Computer Security Foundations Symposium (CSF)

University of Kent

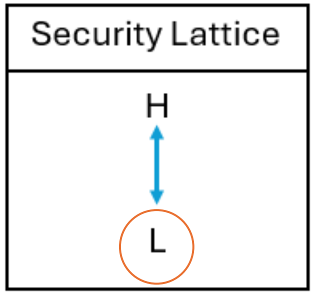
Information Flow Control (IFC)

- IFC – Tracks the flow of information within a computer system, ensuring it adheres to the security policies in effect.
  - Dynamically
  - Statically
    - Non-interference – An attacker cannot differentiate between two computations based on their public values (outputs) when given secret input values.
      - Underlying hyper-property

- Our approach uses a purely static approach
  - Type systems
    - Use security labels or levels to track dependencies between program values
      - Preventing unauthorised data access
      - Adhering to the predefined security properties.

University of Kent

Modal type for classification ($M_l \tau$)



$$\frac{\Phi \; ; \; \Gamma \vdash e : \tau}{\Phi \; ; \; \Gamma \vdash \textbf{return}\, e : M_{\bot}\, \tau} \; \text{T\_RET} \qquad \frac{e \Downarrow_i \Box e' \qquad e' \Downarrow_j v}{\textbf{coret}\; e \Downarrow_{i+j+1} v} \; \text{E\_CORETURN}$$

- The language uses Modal types to track information associated with security labels.
- Classification Modal type - expression that is associated with a security label.

- T-RET : Responsible for lifting an expression into classification modal type (represented with a security label).
- Lowest possible security level in the security lattice.

Security Lattice

H

↕

L

Modal type for classification ($M_l\tau$)

$$\frac{\Phi\ ;\ \Gamma \vdash e_1 : M_l\,\tau \qquad \Phi\ ;\ \Gamma, x : \tau \vdash e_2 : M_{l'}\,\tau' \qquad l \sqsubseteq l'}{\Phi\ ;\ \Gamma \vdash \textbf{bind}\ x \leftarrow e_1\ \textbf{in}\ e_2 : M_{l'}\,\tau'}\ \text{T\_BIND}$$

$$\frac{e \Downarrow_{i_1} v_1 \qquad v_1 \Downarrow^{f}_{i_2} v_1' \qquad e_2[v_1'/x] \Downarrow_{i_3} v_2 \qquad v_2 \Downarrow^{f}_{i_4} v_2'}{\textbf{bind}\ x \leftarrow e_1\ \textbf{in}\ e_2 \Downarrow^{f}_{i_1+i_2+i_3+i_4+1} v_2'}\ \text{EF\_BIND}$$

- Enables sequence computations of the classification modal type and allows the rise of the security level.
- Allows $e_1 : M_l\tau$ the current modal type to be bound to the variable x, resulting in a new expression e_2: $M_{l'}\tau'$ with a higher security level.
- $l \sqsubseteq l'$
  - Bound modal type $M_{l'}\tau'$ has a security level that is same or higher security level from $M_l\tau$.
  - Prevents lower security levels from influencing higher security levels.

Declassification

- Declassification is a procedure of securely releasing information by lowering or eliminating its assigned security level within a system.
- Information flow control mechanisms are then employed to ensure the regulated flow of information based on established security levels and restrictions.
- Guarantees that active attackers may not manipulate the system to learn more secrets than passive attackers already know.
- Split into 4 dimensions [3]:

- **What**: Determines the specific information eligible for declassification.
- **Who**: Identifies the authorised actors permitted to declassify information.
- **Where**: Specifies the locations within the system for declassifying information.
- **When**: Establishes the timing or conditions for information declassification.

[3] Sabelfeld, A. and Sands, D., 2005, June. Dimensions and principles of declassification. In 18th IEEE Computer Security Foundations Workshop (CSFW'05) (pp. 255-269). IEEE.

University of Kent

Logical relations

- Logical relations are used to prove properties about programs in a language.
- Logical relations are proof methods and can be used to prove properties directly.
  - Soundness
  - Termination of well-typed programs.
  - program equivalence
  - Non-interference
  - Hyper properties
  - Security properties e.g. (Confidentiality, integrity)

Base Type value relation – $\mathcal{V}\llparenthesis b \rrparenthesis_\Phi \triangleq \{(n, v) \mid v \in [\![b]\!]\}$

Pairs Type value relation – $\mathcal{V}\llparenthesis \tau_1 \times \tau_2 \rrparenthesis_\Phi \triangleq \{(n, \langle v_1, v_2 \rangle) \mid (n, v_1) \in \mathcal{V}\llparenthesis \tau_1 \rrparenthesis_\Phi \wedge (n, v_2) \in \mathcal{V}\llparenthesis \tau_2 \rrparenthesis_\Phi\}$

Sum Type value relation – $\mathcal{V}\llparenthesis \tau_1 + \tau_2 \rrparenthesis_\Phi \triangleq \{(n, \mathbf{inl}\, v) \mid (n, v) \in \mathcal{V}\llparenthesis \tau_1 \rrparenthesis_\Phi\} \cup \{(n, \mathbf{inr}\, v) \mid n, v \in \mathcal{V}\llparenthesis \tau_2 \rrparenthesis_\Phi\}$

Function Type value relation – $\mathcal{V}\llparenthesis \tau_1 \longrightarrow \tau_2 \rrparenthesis_\Phi \triangleq \{(v, \lambda x.e) \mid \forall j < n, e'.(j, e') \in \mathcal{E}\llparenthesis \tau_1 \rrparenthesis_\Phi \implies (j, e[e'/x]) \in \mathcal{E}\llparenthesis \tau_2 \rrparenthesis_\Phi$

Declassification box Type value relation – $\mathcal{V}\llparenthesis \Box_\phi \tau \rrparenthesis_\Phi \triangleq \{(n, \Box e) \mid \forall j < n(j, e) \in \mathcal{E}\llparenthesis \tau \rrparenthesis_\Phi \wedge$
$\forall j < n.(j, \phi\ e) \in \mathcal{E}\llparenthesis \mathbf{codomain}\ \phi \rrparenthesis_\Phi\}$

Monad classification Type value relation – $\mathcal{V}\llparenthesis M_l \tau \rrparenthesis_\Phi \triangleq \left\{(n, v) \mid \forall i < n.v \Downarrow_i^f v' \implies (n - i, v') \in \mathcal{V}\llparenthesis \tau \rrparenthesis_\Phi\right\}$

Expression relation – $\mathcal{E}\llparenthesis \tau \rrparenthesis_\Phi \triangleq \{(n, e) \mid \forall i < n.e \Downarrow_i v \implies (n - i, v) \in \mathcal{V}\llparenthesis \tau \rrparenthesis_\Phi\}$

Security Lattice

H

$\updownarrow$

L

Declassify box type ($\Box_\phi\, \tau$)

$$\dfrac{\begin{array}{c}\Phi(\phi) = \tau_1 \longrightarrow \tau_2 \\ \Phi\ ;\ \Gamma \vdash e_1 : \Box_\phi\, \tau_1 \\ \Phi\ ;\ \Gamma, x : \tau_2 \vdash e_2 : \tau'\end{array}}{\Phi\ ;\ \Gamma \vdash \mathbf{dec}\, x \leftarrow \phi\ e_1\ \mathbf{in}\ e_2 : \tau'}\ \text{T\_DEC}$$

$$\dfrac{\begin{array}{c}e_1 \Downarrow_i \Box\, e \\ e_2[\phi\, e/x] \Downarrow_j v'\end{array}}{\mathbf{dec}\, x \leftarrow \phi\ e_1\ \mathbf{in}\ e_2 \Downarrow_{i+j+1} v'}\ \text{E\_DEC}$$

- T-dec process by which classified information is lowered from a high-security level to a lower-security level.
- ϕ is a declassification policy, a function that is not syntactically typed checked.
    - Semantically typed checked in the model.
- Our logical relation for declassification in Unary, for simplicity:

T-EXTRACT/Co-return($\Box_\phi \tau$)

$$\frac{\Phi; \Gamma \vdash e : \Box_\phi \tau}{\Phi; \Gamma \vdash \textbf{coreturn}\ e : \tau}\ \text{T\_CORETURN} \qquad \frac{e \Downarrow_i \Box e' \qquad e' \Downarrow_j v}{\textbf{coret}\ e \Downarrow_{i+j+1} v}\ \text{E\_CORETURN}$$

- Extract is used to extract values of the box type box-type.
- Doesn't provide a means to inject value back into the box-type.

T-SPLIT/Cojoin ($\Box_\phi \tau$)

$$\frac{\Phi;\Gamma \vdash e : \Box_\phi \tau \qquad \phi = \phi_2.\phi_1 \qquad \Phi;\Gamma \vdash \phi_1 : \tau \longrightarrow \tau'}{\Phi;\Gamma \vdash \textbf{Split} : \Box_{\phi'} \Box_{\phi_1} \tau} \text{ E-SPLIT}$$

Where $\phi' = \lambda x.\mathbf{dec}\ y = \phi_1\ x\ \mathbf{in}\ \phi_2\ y$

$$\frac{e \Downarrow_i \Box e'}{\textbf{Split}\ \ e \Downarrow_{i+1} \Box\Box\, e'} \text{ E-SPLIT}$$

- ϕ is split into ϕ_1 and ϕ_2
- Adds declassification box type $\Box e$ in context of another $\Box e$ making it $\Box\Box e$.
- ϕ_1 is typed check.
- ϕ_2 isn't typed check only semantically checked in the model.
- Leakage of information:
  - possible in ϕ_2
  - Not possible in ϕ_1

Where $\phi' = \lambda x.\mathbf{dec}\ y = \phi_1\ x\ \mathbf{in}\ \phi_2\ y$

- ensures that ϕ_1 is properly typed and that $\phi_2$provides an additional layer of box type.

University of Kent

Three co-monadic laws

Co-monadic laws	Our version	Haskell version
Co-monadic law 1:	**Co-monadic law 1-** $\forall i.(i, \lambda x.\mathit{Extract/co\text{-}return}(\mathit{Split/cojoin}\ x) \in \mathcal{E}(\!\| \Box_\phi \tau \longrightarrow \Box_\phi \tau \|\!)_\Phi \iff \forall i.(i, \mathit{id}) \in \mathcal{E}(\!\| \Box_\phi \tau \longrightarrow \Box_\phi \tau \|\!)_\Phi$	extract.duplicate = id
Co–monadic law 2:	**Co-monadic law 2-** $\forall i.(i, \lambda x.\mathit{fmapD}\ \mathit{Extract/Coreturn}(\mathit{Split/Cojoin}\ x) \in \mathcal{E}(\!\| \Box_\phi \tau \longrightarrow \Box_\phi \tau \|\!)_\Phi \iff \forall i.(i, \mathit{id}) \in \mathcal{E}(\!\| \Box_\phi \tau \longrightarrow \Box_\phi \tau \|\!)_\Phi$	fmap extract.duplicate = id
Co-monadic law 3:	**Co-monadic law 3-** $(\forall i.(i, \lambda x.\mathbf{Cojoin}.(\mathbf{Cojoin}\ x)) \in \mathcal{E}(\!\| \Box_\phi \tau \longrightarrow \Box_{\mathbf{id}} \Box_{\mathbf{id}} \Box_\phi \tau \|\!)_\Phi \iff \forall i.(i, \lambda x.\mathbf{fmap_D}\ \mathbf{cojoin}.(\mathbf{cojoin}\ x)) \in \mathcal{E}(\!\| \Box_\phi \tau \longrightarrow \Box_{\mathbf{id}} \Box_{\mathbf{id}} \Box_\phi \tau \|\!)_\Phi)$.	duplicate.duplicate = fmap duplicate.duplicate

- Our proofs done for co-mandic laws establish that both sides of the theorem are logically equivalent by expressing them as bi-implications using logical relations.

Soundness and Model

(Soundness).

$$
\begin{aligned}
&.\forall v_1, v_2, e, i. \\
&\phi : M_{\mathrm{H}} Bool \to M_{\mathrm{H}} Bool; \\
&x : \Box_\phi M_{\mathrm{H}} Bool \vdash \textbf{dec y} = \phi(x)e : M_{\mathrm{L}} Bool \wedge \\
&.;. \vdash \Box v_1 : \Box_\phi M_{\mathrm{H}} Bool \wedge \\
&.;. \vdash \Box v_2 : \Box_\phi M_{\mathrm{H}} Bool \wedge \\
&\mu\ v_1 \Downarrow^f v' \wedge \mu\ v_2 \Downarrow^f v' \wedge \\
&(\textbf{dec y} = \phi(x)e)[v_1/x][\mu/\phi] \Downarrow^f_- v_1' \wedge \\
&(\textbf{dec y} = \phi(x)e)[v_2/x][\mu/\phi] \Downarrow^f_- v_2' \implies v_1' = v_2'
\end{aligned}
$$

- Since our language incorporates safe information leakage and novel security policies, it is vital to compare two runs of the system to ensure consistent behaviour and prevent unintended information leaks.
- This necessity highlights the importance of using both unary and binary logical relations.

(Unary fundamental theorem). $\forall \Gamma, e, \tau, \delta, n.$

$$\Phi;\Gamma \vdash e : \tau \wedge (n, \delta) \in \mathcal{G}[\![\Gamma]\!]_\Phi \wedge (n, \mu) \in \mathcal{G}[\![\Phi]\!] \implies (n, e\ \delta\ \mu) \in \mathcal{E}(\!|\tau\ \mu|\!)_\Phi$$

(Binary fundamental theorem). $\forall \Gamma, A, e, \tau, \gamma, n.$

$$
\begin{aligned}
&\Phi\ ;\ \Gamma \vdash e : \tau \wedge (n, \gamma) \in \mathcal{G}[\![\Gamma]\!]^{\mathcal{A}}_\Phi \wedge (n, \nu) \in \mathcal{G}[\![\Phi]\!]^{\mathcal{A}}_\Phi \implies \\
&(n, e(\gamma \downarrow_1)(\nu \downarrow_1, e(\gamma \downarrow_2)(\nu \downarrow_2)) \in \mathcal{E}[\![\tau\nu \downarrow_1]\!]^{\mathcal{A}}_\Phi
\end{aligned}
$$

- The unary and Binary Fundamental Theorems ensure the correctness of typing rules and their semantics through the corresponding logical relations.

University of Kent

Logical Relations

(Unary logical relation).

$$
\begin{array}{lcl}
\mathcal{V}\llparenthesis\, b \,\rrparenthesis_\Phi & \triangleq & \{(n, v) \mid v \in [\![b]\!]\} \\[1ex]
\mathcal{V}\llparenthesis\, \tau_1 \times \tau_2 \,\rrparenthesis_\Phi & \triangleq & \{(n, \langle v_1, v_2\rangle) \mid (n, v_1) \in \mathcal{V}\llparenthesis\, \tau_1 \,\rrparenthesis_\Phi \wedge (n, v_2) \in \mathcal{V}\llparenthesis\, \tau_2 \,\rrparenthesis_\Phi\} \\[1ex]
\mathcal{V}\llparenthesis\, \tau_1 + \tau_2 \,\rrparenthesis_\Phi & \triangleq & \{(n, \mathbf{inl}\ v) \mid (n, v) \in \mathcal{V}\llparenthesis\, \tau_1 \,\rrparenthesis_\Phi\} \cup \{(n, \mathbf{inr}\ v) \mid n, v \in \mathcal{V}\llparenthesis\, \tau_2 \,\rrparenthesis_\Phi\} \\[1ex]
\mathcal{V}\llparenthesis\, \tau_1 \longrightarrow \tau_2 \,\rrparenthesis_\Phi & \triangleq & \{(v, \lambda x.e) \mid \forall j < n, e'.(j, e') \in \mathcal{E}\llparenthesis\, \tau_1 \,\rrparenthesis_\Phi \implies (j, e[{}^{e'}\!/\!_x]) \in \mathcal{E}\llparenthesis\, \tau_2 \,\rrparenthesis_\Phi\} \\[1ex]
\mathcal{V}\llparenthesis\, \Box_\phi\, \tau \,\rrparenthesis_\Phi & \triangleq & \{(n, \Box e) \mid (n, e) \in \mathcal{V}\llparenthesis\, \tau \,\rrparenthesis_\Phi \wedge \\
 & & \forall i < n.(i, \phi\ e) \in \mathcal{V}\llparenthesis\, \mathbf{codomain}\ \phi \,\rrparenthesis_\Phi\} \\[1ex]
\mathcal{V}\llparenthesis\, M_l\, \tau \,\rrparenthesis_\Phi & \triangleq & \left\{(n, v) \mid \forall i < n.v \Downarrow_i^f v' \implies (n - i, v') \in \mathcal{V}\llparenthesis\, \tau \,\rrparenthesis_\Phi\right\} \\[1ex]
\mathcal{E}\llparenthesis\, \tau \,\rrparenthesis_\Phi & \triangleq & \{(n, e) \mid \forall i < n.e \Downarrow_i v \implies (n - i, v) \in \mathcal{V}\llparenthesis\, \tau \,\rrparenthesis_\Phi\}
\end{array}
$$

(Binary logical relation).

$$
\begin{array}{lcl}
\mathcal{V}[\![b]\!]_\Phi^\mathcal{A} & \triangleq & \{(n, v_1, v_2) \mid v_1 = v_2 \wedge \{v_1, v_2\} \in [\![b]\!]\} \\[1ex]
\mathcal{V}[\![\tau_1 \times \tau_2]\!]_\Phi^\mathcal{A} & \triangleq & \left\{(n, \langle v_1, v_2\rangle, \langle v_1', v_2'\rangle) \mid (n, v_1, v_1') \in \mathcal{V}[\![\tau_1]\!]_\Phi^\mathcal{A} \wedge (n, v_2, v_2') \in \mathcal{V}[\![\tau_2]\!]_\Phi^\mathcal{A}\right\} \\[1ex]
\mathcal{V}[\![\tau_1 + \tau_2]\!]_\Phi^\mathcal{A} & \triangleq & \left\{(n, \mathbf{inl}\ v, \mathbf{inl}\ v') \mid (n, v, v') \in \mathcal{V}[\![\tau_1]\!]_\Phi^\mathcal{A}\right\} \cup \\
 & & \left\{(n, \mathbf{inr}\ v, \mathbf{inr}\ v') \mid (n, v, v') \in \mathcal{V}[\![\tau_2]\!]_\Phi^\mathcal{A}\right\} \\[1ex]
\mathcal{V}[\![\tau_1 \longrightarrow \tau_2]\!]_\Phi^\mathcal{A} & \triangleq & \{(n, \lambda x.e_1, \lambda x.e_2) \mid \forall j < n, e_1, e_2.((j, e_1, e_2 \in \mathcal{E}[\![\tau_1]\!]_\Phi^\mathcal{A} \implies \\
 & & (j, e_1[{}^{e_1}\!/\!_x], e_2[{}^{e_2}\!/\!_x]) \in \mathcal{E}[\![\tau_2]\!]_\Phi^\mathcal{A}) \wedge \\
 & & \forall j, e'.((j, e') \in \mathcal{E}[\![\tau_1]\!]_\Phi \implies (j, e_1[{}^{e'}\!/\!_x]) \in \mathcal{E}[\![\tau_2]\!]_\Phi) \wedge \\
 & & \forall j, e'.((j, e') \in \mathcal{E}[\![\tau_1]\!]_\Phi \implies (j, e_2[{}^{e'}\!/\!_x]) \in \mathcal{E}[\![\tau_2]\!]_\Phi)\} \\[1ex]
\mathcal{V}[\![\Box_\phi\, \tau]\!]_\Phi^\mathcal{A} & \triangleq & \{(n, \Box e, \Box e') \mid (n, e, e') \in \mathcal{V}[\![\tau]\!]_\Phi^\mathcal{A} \wedge \\
 & & \forall i < n.(i, \phi\ e, \phi\ e') \in \mathcal{V}[\![\mathbf{codomain}\ \phi]\!]_\Phi^\mathcal{A}\} \\[1ex]
\mathcal{V}[\![M_l\, \tau]\!]_\Phi^\mathcal{A} & \triangleq & \left\{(n, v_1, v_2) \mid \forall i < n, v_1', v_2'.v_1 \Downarrow_i^f v_1' \wedge v_2 \Downarrow_i^f v_2' \implies ValEq(A, \Phi, l, n - i, v_1', v_2', \tau\right\} \\[1ex]
\mathcal{E}[\![\tau]\!]_\Phi^\mathcal{A} & \triangleq & \left\{(n, e_1, e_2) \mid \forall i < n.e_1 \Downarrow_i v_1 \wedge e_2 \Downarrow v_2 \implies (n - i, v_1, v_2) \in \mathcal{V}[\![\tau]\!]_\Phi^\mathcal{A}\right\}
\end{array}
$$

Term and Policy logical relations

(Unary interpretation of term and policy context).

$$\mathcal{G}[\![\Gamma]\!]_\Phi \triangleq \{(n,\delta) \mid (\mathrm{dom}(\Gamma) \subseteq \mathrm{dom}(\delta) \wedge \forall x \in \mathrm{dom}(\Gamma).(n,\delta,x) \in \mathcal{E}(\!|\Gamma(x)|\!)_\Phi\}$$

$$\mathcal{G}[\![\Phi]\!] \triangleq \{(n,\mu) \mid (\mathrm{dom}(\Phi) \subseteq \mathrm{dom}(\mu) \wedge \forall \phi \in \mathrm{dom}(\Phi) \wedge \forall i < n.(i,\mu,\phi) \in \mathcal{V}(\!|\Phi(\phi)|\!)_\Phi\}$$

(Binary interpretation of term and policy context).

$$\mathcal{G}[\![\Gamma]\!]^A_\Phi \triangleq \left\{(n,\gamma) \mid \mathrm{dom}(\Gamma) \subseteq \mathrm{dom}(\gamma) \wedge \forall x \in \mathrm{dom}(\Gamma).(n,\pi_1(\gamma(x)),\pi_2(\gamma(x))) \in \mathcal{E}[\![\Gamma(x)]\!]^A_\Phi\right\}$$

$$\begin{aligned}\mathcal{G}[\![\Phi]\!]^A \triangleq\ & \{(n,\nu) | \mathrm{dom}(\Phi) \subseteq \mathrm{dom}(\nu) \wedge \forall \phi \in \mathrm{dom}(\Phi).\pi_1\ (\nu\ \phi) = \pi_2\ (\nu\ \phi) \wedge \\ & \forall i < n.(i,\pi_1(\nu\ \phi),\pi_2(\nu\ \phi)) \in \mathcal{V}(\!|\Phi(\phi)|\!)_\Phi\}\end{aligned}$$

University of Kent

Questions