# 機密度パラメータ付き情報流解析のための型検査アルゴリズムとJavaアノテーション

**Type Checking Algorithm and Java Annotations for Information Flow Analysis with Parameterized Secrecy**

**桑原 寛明**[*] **國枝 義敏**[†]

**Summary.** This paper proposes a type checking algorithm and Java annotations toward an implementation of information flow analysis with parameterized secrecy. This algorithm generates two constraint sets for secrecy of each program constructs and checks the satisfiability of these constraint sets. If both constraint sets are satisfiable, type checking succeeds and it means there are no illegal information flow. We extend existing Java annotations for information flow analysis to support parameterized secrecy and show a simple example of annotated program.

## 1 はじめに

プログラムの静的解析により機密情報が外部に漏れないことを検査する手法として，型検査に基づく情報流解析が提案されている [1] [2] [3] [4]．型検査に基づく情報流解析では，データの機密度を型として利用し，型付け可能なプログラムが非干渉性を満たすように型システムを構築する．非干渉性は，機密度の低いデータが機密度の高いデータに直接および間接的に依存しないことを表し，機密データ自体に加え機密データを推測できる情報も漏らさないという意味でよい性質である．

型検査に基づく情報流解析では，プログラム中の変数や関数の返り値の型として機密度を指定する必要がある．この時，通常は具体的な機密度を指定する．そのため，汎用的なコレクションフレームワークのような扱うデータの機密度を作成時には決められないプログラムの場合，指定される機密度のみが異なるクローンを複数作成することになり望ましくない．この問題に対し，機密度のパラメータ化が提案されている [5]．この手法では，Java 言語におけるジェネリックなクラスのように，具体的な機密度の代わりに機密度パラメータを用いてクラスを定義し，インスタンス生成時に具体的な機密度を機密度パラメータに割り当てる．これにより，機密度のみが異なるクローンの作成を回避できる．機密度パラメータに対応した情報流解析のための型システムも提案されており，型付け可能なプログラムは機密度パラメータにどのような機密度を割り当てても非干渉性を満たすことが保証されている．

[5] の型システムに基づいて情報流解析を実現するためには，型検査アルゴリズムの構築と，機密度パラメータをプログラム中に記述する手段の提供が必要である．機密度パラメータのない従来の情報流解析に対しては，制約集合の充足問題に帰着される型検査アルゴリズム [6] と，機密度をプログラム中に記述するための Java アノテーション [7] が提案されている．構文を拡張するのではなく，構文の一部であるアノテーションを活用することで，既存の開発ツールの利用を継続できる．

本稿では，機密度パラメータに対応した情報流解析のための型検査アルゴリズムと，機密度パラメータを Java プログラム中に記述するためのアノテーションを提案する．型検査アルゴリズムについては，型付け規則における機密度に関する制約を機密度定数に関する制約と機密度パラメータに関する制約に分割することで，既存手法と同様に型検査を制約集合の充足問題に帰着させることで実現する．Java アノテーションについては，機密度パラメータを扱えるように既存のアノテーション [7] を拡張すると同時に，機密度パラメータに対する機密度の割り当てを記述するため

[*]Hiroaki Kuwabara, 南山大学情報センター

[†]Yoshitoshi Kunieda, 立命館大学情報理工学部

$$
\begin{array}{rl}
T ::= & \mathrm{Bool} \mid \mathrm{Null} \mid C \qquad \tau ::= (T\!<\!\overline{\rho}\!>, \rho) \qquad \rho ::= (\eta, \mathcal{X}) \\
CL ::= & \mathrm{class}\ C\!<\!\overline{X}\!>\ \rho\ \mathrm{extends}\ C\!<\!\overline{\rho}\!>\ \{\overline{\tau\ f;}\ \overline{M}\} \\
M ::= & \tau\ m(\overline{\tau\ x})\ \rho\ B \qquad B ::= \{\overline{\tau\ x;}\ \overline{S;}\} \\
S ::= & x\ =\ e \mid e.f = e \mid \mathrm{if}\ (e)\ B\ \mathrm{else}\ B \mid x\ =\ e.m(\overline{e}) \mid x\ =\ \mathrm{new}\ C\!<\!\overline{\rho}\!> \\
e ::= & x \mid e.f \mid \mathrm{true} \mid \mathrm{false} \mid \mathrm{null} \mid \mathrm{this} \mid e\ ==\ e
\end{array}
$$

**図 1　$OO_G$ の構文**

のアノテーションを新たに定義する．

## 2　機密度パラメータと情報流解析

機密度をパラメータに取るクラス定義を導入したオブジェクト指向言語 $OO_G$ と，$OO_G$ プログラムに対する情報流解析のための型システムを [5] に従って示す．以下では，機密度定数の束 $(\mathcal{H}, \sqsubset)$ と機密度パラメータ全体の集合 $\mathcal{Y}$ を仮定する．束の最小元を $\bot_{\mathcal{H}}$ と表す．機密度 $\rho$ を機密度定数 $\eta \in \mathcal{H}$ と機密度パラメータの集合 $\mathcal{X} \subseteq \mathcal{Y}$ の組 $(\eta, \mathcal{X})$ として定義する．直観的には，$(\eta, \mathcal{X})$ は $\eta$ とすべての $X \in \mathcal{X}$ の結びである．機密度の集合 $\mathcal{P} = \mathcal{H} \times 2^{\mathcal{Y}}$ 上に機密度の大小関係 $\preceq$ を $(\eta_1, \mathcal{X}_1) \preceq (\eta_2, \mathcal{X}_2)$ **iff** $\eta_1 \sqsubset \eta_2 \wedge \mathcal{X}_1 \subseteq \mathcal{X}_2$ と定義する．この時，$(\mathcal{P}, \preceq)$ は束である．

$OO_G$ の構文を図 1 に示す．型情報 $\tau$ はデータ型 $T\!<\!\overline{\rho}\!>$ と機密度 $\rho$ の組である．$T\!<\!\overline{\rho}\!>$ は $T$ の機密度パラメータ $\overline{X}$ に機密度 $\overline{\rho}$ を割り当てた型である．$T$ が機密度パラメータを持たず $\overline{X}$ の長さが 0 の場合は単に $T$ と書く．機密度 $(\eta, \mathcal{X})$ について，機密度パラメータの集合 $\mathcal{X}$ が空の時は単に $\eta$ と書き，機密度定数 $\eta$ が $\bot_{\mathcal{H}}$ かつ $\mathcal{X}$ が $\{X\}$ の時は単に $X$ と書く．クラス定義 $CL$ は，クラス名 $C$ と，いずれも 0 個以上の機密度パラメータの宣言 $\overline{X}$，フィールドの宣言 $\overline{\tau\ f;}$，メソッド定義 $\overline{M}$ からなる．

$OO_G$ の型付け規則を図 2 に示す．$\mathit{glevel}(T\!<\!\overline{\rho}\!>)$ は $T\!<\!\overline{\rho}\!>$ の機密度，$\mathit{gsfields}(T\!<\!\overline{\rho}\!>)$ は $T\!<\!\overline{\rho}\!>$ が持つフィールドの名前と型の集合，$\mathit{gsmethod}(m, T\!<\!\overline{\rho}\!>)$ は $T\!<\!\overline{\rho}\!>$ が持つメソッド $m$ の型を表す．$\trianglelefteq$ はサブタイプ関係を表す．文とブロックの型判定式 $\Delta \vdash S : (\rho_s, \rho_h)$ は，$S$ で代入される変数の機密度が $\rho_s$ 以上，フィールドへの代入などで変更されるヒープ上のデータの機密度が $\rho_h$ 以上であることを表す．式の型判定式 $\Delta \vdash e : (T\!<\!\overline{\rho}\!>, \rho)$ は，$e$ のデータ型が $T\!<\!\overline{\rho}\!>$ かそのサブタイプ，機密度が $\rho$ 以下であることを表す．

## 3　型検査アルゴリズム

型検査は，(1) プログラムが満たすべき機密度に関する制約集合を求め，(2) 得られた制約集合の充足可能性を判定する，という手順で行う．制約集合が充足可能であれば型検査に成功，充足不能であれば失敗と判定する．CDEC 規則より，型検査はメソッド宣言ごとに独立して実施できる．簡単のため，データ型については正しく型付けされていることを前提とする．

制約集合の生成規則を図 3 に示す．図中の $\kappa$ や $\kappa_s$ などは機密度定数，$\nu$ や $\nu_s$ などは機密度パラメータの集合を表す変数である．プログラム中の文や式の機密度を，機密度定数および機密度パラメータの集合を表す新しい 2 つの変数から構成し，図 2 の型付け規則に従って機密度のうち機密度定数に関する制約集合と機密度パラメータの集合に関する制約集合を別々に生成する．$\Delta \Vdash S : ((\kappa_s, \nu_s), (\kappa_h, \nu_h)) \parallel \mathsf{C}; \mathsf{D}$ は，文あるいはブロック $S$ が型環境 $\Delta$ の下で型付け可能であるための機密度定数と機密度パラメータの集合に関する制約集合がそれぞれ $\mathsf{C}$ と $\mathsf{D}$ であることを表す．式についても同様である．メソッド定義に C-MDEC 規則を適用して得られる制約集合 $\mathsf{C}$ と $\mathsf{D}$ がともに充足可能であればそのメソッド定義は型検査に成功する．紙数の制限により証明は省くが，あるメソッド定義について，制約集合が充足可能であれば解に基づいて型付け可能であり，型付け可能であれば制約集合は充足可能である．

$$\frac{\begin{array}{l} \mathit{glevel}(D\overline{<\rho>}) \preceq \rho \quad C\overline{<X>}\ \rho \text{ extends } D\overline{<\rho>} \vdash M \text{ for each } M \in \overline{M} \\ \mathit{glevel}(D\overline{<\rho>}) \neq \rho \Rightarrow \\ \qquad \text{every } m \text{ with } \mathit{gsmethod}(m, D\overline{<\rho>}) \text{ defined is overridden in } C \end{array}}{\vdash C\overline{<X>}\ \rho \text{ extends } D\overline{<\rho>}\ \{\overline{\tau\ f};\ \overline{M}\}}\ [\text{CDEC}]$$

$$\frac{\begin{array}{l} \overline{x : \tau_x},\ \text{this} : (C\overline{<X>}, \rho),\ \mathit{result} : \tau_r \vdash B : (\rho_s, \rho'_h) \\ \mathit{gsmethod}(m, D\overline{<\rho>}) \text{ is defined } \Rightarrow \mathit{gsmethod}(m, D\overline{<\rho>}) = \overline{x : \tau_x} \xrightarrow{\rho_h} \tau_r \\ \rho_h \preceq \rho'_h \end{array}}{C\overline{<X>}\ \rho \text{ extends } D\overline{<\rho>} \vdash \tau_r\ m(\overline{\tau_x\ x})\ \rho_h\ B}\ [\text{MDEC}]$$

$$\frac{\Delta,\ \overline{x : (T_x\overline{<\rho_x>}, \rho_x)} \vdash S_i : (\rho^i_s, \rho^i_h) \quad \rho_s \preceq \rho^i_s \quad \rho_h \preceq \rho^i_h \quad i \in \{1, \ldots, n\}}{\Delta \vdash \{\overline{(T_x\overline{<\rho_x>}, \rho_x)\ x};\ S_1; \ldots S_n;\} : (\rho_s, \rho_h)}\ [\text{BLOCK}]$$

$$\frac{\begin{array}{l} \Delta \vdash x : (T_x\overline{<\rho_x>}, \rho_x) \quad \Delta \vdash e : (T_e\overline{<\rho_e>}, \rho_e) \\ T_e\overline{<\rho_e>} \trianglelefteq T_x\overline{<\rho_x>} \quad \rho_e \preceq \rho_x \quad \rho_s \preceq \rho_x \end{array}}{\Delta \vdash x = e : (\rho_s, \rho_h)}\ [\text{ASSIGN1}]$$

$$\frac{\begin{array}{l} \Delta \vdash e_1 : (T_1\overline{<\rho_1>}, \rho_1) \quad \Delta \vdash e_2 : (T_2\overline{<\rho_2>}, \rho_2) \\ f : (T_f\overline{<\rho_f>}, \rho_f) \in \mathit{gsfields}(T_1\overline{<\rho_1>}) \\ T_2\overline{<\rho_2>} \trianglelefteq T_f\overline{<\rho_f>} \quad \rho_1 \preceq \rho_f \quad \rho_2 \preceq \rho_f \quad \rho_h \preceq \rho_f \end{array}}{\Delta \vdash e_1.f = e_2 : (\rho_s, \rho_h)}\ [\text{ASSIGN2}]$$

$$\frac{\begin{array}{l} \Delta \vdash x : (T_x\overline{<\rho_x>}, \rho_x) \quad \Delta \vdash e : (T_e\overline{<\rho_e>}, \rho_e) \\ y_1 : (T'_1\overline{<\rho'_1>}, \rho'_1), \ldots, y_n : (T'_n\overline{<\rho'_n>}, \rho'_n) \xrightarrow{\rho'_h} (T_r\overline{<\rho_r>}, \rho_r) \\ \qquad = \mathit{gsmethod}(m, T_e\overline{<\rho_e>}) \\ \Delta \vdash e_i : (T_i\overline{<\rho_i>}, \rho_i) \quad T_i\overline{<\rho_i>} \trianglelefteq T'_i\overline{<\rho'_i>} \quad \rho_i \preceq \rho'_i \quad i \in \{1, \ldots, n\} \\ T_r\overline{<\rho_r>} \trianglelefteq T_x\overline{<\rho_x>} \quad \rho_e \preceq \rho_x \quad \rho_r \preceq \rho_x \quad \rho_s \preceq \rho_x \quad \rho_e \preceq \rho'_h \quad \rho_h \preceq \rho'_h \end{array}}{\Delta \vdash x = e.m(e_1, \ldots, e_n) : (\rho_s, \rho_h)}\ [\text{CALL}]$$

$$\frac{\begin{array}{l} \Delta \vdash x : (T_x\overline{<\rho_x>}, \rho_x) \quad C\overline{<\rho_C>} \trianglelefteq T_x\overline{<\rho_x>} \\ \mathit{glevel}(C\overline{<\rho_C>}) \preceq \rho_x \quad \rho_s \preceq \rho_x \quad \rho_h \preceq \mathit{glevel}(C\overline{<\rho_C>}) \end{array}}{\Delta \vdash x = \text{new } C\overline{<\rho_C>} : (\rho_s, \rho_h)}\ [\text{NEW}]$$

$$\frac{\begin{array}{l} \Delta \vdash e : (\text{Bool}, \rho_e) \quad \Delta \vdash B_t : (\rho^t_s, \rho^t_h) \quad \Delta \vdash B_f : (\rho^f_s, \rho^f_h) \\ \rho_e \preceq \rho_s \quad \rho_s \preceq \rho^t_s \quad \rho_s \preceq \rho^f_s \quad \rho_e \preceq \rho_h \quad \rho_h \preceq \rho^t_h \quad \rho_h \preceq \rho^f_h \end{array}}{\Delta \vdash \text{if } (e)\ B_t \text{ else } B_f : (\rho_s, \rho_h)}\ [\text{IF}]$$

$$\frac{(T\overline{<\rho>}, \rho) = \Delta(x)}{\Delta \vdash x : (T\overline{<\rho>}, \rho)}\ [\text{VAR}] \qquad \frac{\begin{array}{l} \Delta \vdash e : (T_e\overline{<\rho_e>}, \rho_e) \\ f : (T_f\overline{<\rho_f>}, \rho_f) \in \mathit{gsfields}(T_e\overline{<\rho_e>}) \\ \rho_e \preceq \rho \quad \rho_f \preceq \rho \end{array}}{\Delta \vdash e.f : (T_f\overline{<\rho_f>}, \rho)}\ [\text{FIELD}]$$

$$\frac{c \in \{\text{true}, \text{false}\}}{\Delta \vdash c : (\text{Bool}, \bot_{\mathcal{H}})}\ [\text{BOOL}] \qquad \frac{}{\Delta \vdash \text{null} : (\text{Null}, \bot_{\mathcal{H}})}\ [\text{NULL}]$$

$$\frac{(T\overline{<\rho>}, \rho) = \Delta(\text{this})}{\Delta \vdash \text{this} : (T\overline{<\rho>}, \rho)}\ [\text{THIS}] \qquad \frac{\begin{array}{l} \Delta \vdash e_1 : (T\overline{<\rho>}, \rho_1) \quad \Delta \vdash e_2 : (T\overline{<\rho>}, \rho_2) \\ \rho_1 \preceq \rho \quad \rho_2 \preceq \rho \end{array}}{\Delta \vdash e_1 \ == \ e_2 : (\text{Bool}, \rho)}\ [\text{EQ}]$$

**図 2　型付け規則**

$$\frac{\overline{x:\tau_x},\ \text{this} : (C\!<\!\overline{X}\!>, \rho),\ result : \tau_r \Vdash B : ((\kappa_s, \nu_s), (\kappa_h, \nu_h)) \parallel \mathsf{C_B}; \mathsf{D_B} \quad \mathsf{C} = \mathsf{C_B} \cup \{\eta \sqsubseteq \kappa_h\} \quad \mathsf{D} = \mathsf{D_B} \cup \{\mathcal{X} \subseteq \nu_h\}}{C\!<\!\overline{X}\!>\ \rho \text{ extends } D\!<\!\overline{\rho}\!> \Vdash \tau_r\ m(\overline{\tau_x\ x})\ (\eta, \mathcal{X})\ B \parallel \mathsf{C}; \mathsf{D}}\ \text{[C-MDEC]}$$

$$\frac{\Delta,\ \overline{x : (T_x\!<\!\overline{\rho_x}\!>, \rho_x)} \Vdash S_i : ((\kappa_s^i, \nu_s^i), (\kappa_h^i, \nu_h^i)) \parallel \mathsf{C}_i; \mathsf{D}_i \quad i \in \{1, \ldots, n\} \quad \mathsf{C} = \bigcup_i (\mathsf{C}_i \cup \{\kappa_s \sqsubseteq \kappa_s^i, \kappa_h \sqsubseteq \kappa_h^i\}) \quad \mathsf{D} = \bigcup_i (\mathsf{D}_i \cup \{\nu_s \subseteq \nu_s^i, \nu_h \subseteq \nu_h^i\})}{\Delta \Vdash \{\overline{(T_x\!<\!\overline{\rho_x}\!>, \rho_x)\ x};\ S_1; \ldots S_n;\} : ((\kappa_s, \nu_s), (\kappa_h, \nu_h)) \parallel \mathsf{C}; \mathsf{D}}\ \text{[C-BLOCK]}$$

$$\frac{\Delta \Vdash x : (T_x\!<\!\overline{\rho_x}\!>, (\kappa_x, \nu_x)) \parallel \mathsf{C_x}; \mathsf{D_x} \quad \Delta \Vdash e : (T_e\!<\!\overline{\rho_e}\!>, (\kappa_e, \nu_e)) \parallel \mathsf{C_e}; \mathsf{D_e} \quad \mathsf{C} = \mathsf{C_x} \cup \mathsf{C_e} \cup \{\kappa_e \sqsubseteq \kappa_x, \kappa_s \sqsubseteq \kappa_x\} \quad \mathsf{D} = \mathsf{D_x} \cup \mathsf{D_e} \cup \{\nu_e \subseteq \nu_x, \nu_s \subseteq \nu_x\}}{\Delta \Vdash x = e : ((\kappa_s, \nu_s), (\kappa_h, \nu_h)) \parallel \mathsf{C}; \mathsf{D}}\ \text{[C-ASSIGN1]}$$

$$\frac{\Delta \Vdash e_1 : (T_1\!<\!\overline{\rho_1}\!>, (\kappa_1, \nu_1)) \parallel \mathsf{C_1}; \mathsf{D_1} \quad \Delta \Vdash e_2 : (T_2\!<\!\overline{\rho_2}\!>, (\kappa_2, \nu_2)) \parallel \mathsf{C_2}; \mathsf{D_2} \quad f : (T_f\!<\!\overline{\rho_f}\!>, (\eta, \mathcal{X})) \in \mathit{gsfields}(T_1\!<\!\overline{\rho_1}\!>) \quad \mathsf{C} = \mathsf{C_1} \cup \mathsf{C_2} \cup \{\kappa_1 \sqsubseteq \eta, \kappa_2 \sqsubseteq \eta, \kappa_h \sqsubseteq \eta\} \quad \mathsf{D} = \mathsf{D_1} \cup \mathsf{D_2} \cup \{\nu_1 \subseteq \mathcal{X}, \nu_2 \subseteq \mathcal{X}, \nu_h \subseteq \mathcal{X}\}}{\Delta \Vdash e_1.f = e_2 : ((\kappa_s, \nu_s), (\kappa_h, \nu_h)) \parallel \mathsf{C}; \mathsf{D}}\ \text{[C-ASSIGN2]}$$

$$\frac{\begin{array}{l}\Delta \Vdash x : (T_x\!<\!\overline{\rho_x}\!>, (\kappa_x, \nu_x)) \parallel \mathsf{C_x}; \mathsf{D_x} \quad \Delta \Vdash e : (T_e\!<\!\overline{\rho_e}\!>, (\kappa_e, \nu_e)) \parallel \mathsf{C_e}; \mathsf{D_e} \\ y_1 : (T_1'\!<\!\overline{\rho_1'}\!>, (\eta_1, \mathcal{X}_1)), \ldots, y_n : (T_n'\!<\!\overline{\rho_n'}\!>, (\eta_n, \mathcal{X}_n)) \\ \quad \xrightarrow{(\eta_h, \mathcal{X}_h)} (T_r\!<\!\overline{\rho_r}\!>, (\eta_r, \mathcal{X}_r)) = \mathit{gsmethod}(m, T_e\!<\!\overline{\rho_e}\!>) \\ \Delta \Vdash e_i : (T_i\!<\!\overline{\rho_i}\!>, (\kappa_i, \nu_i)) \parallel \mathsf{C}_i; \mathsf{D}_i \quad i \in \{1, \ldots, n\} \\ \mathsf{C} = \mathsf{C_x} \cup \mathsf{C_e} \cup \bigcup_i (\mathsf{C}_i \cup \{\kappa_i \sqsubseteq \eta_i\}) \cup \{\kappa_e \sqsubseteq \kappa_x, \eta_r \sqsubseteq \kappa_x, \kappa_s \sqsubseteq \kappa_x, \kappa_e \sqsubseteq \eta_h, \kappa_h \sqsubseteq \eta_h\} \\ \mathsf{D} = \mathsf{D_x} \cup \mathsf{D_e} \cup \bigcup_i (\mathsf{D}_i \cup \{\nu_i \subseteq \mathcal{X}_i\}) \cup \{\nu_e \subseteq \nu_x, \mathcal{X}_r \subseteq \nu_x, \nu_s \subseteq \nu_x, \nu_e \subseteq \mathcal{X}_h, \nu_h \subseteq \mathcal{X}_h\}\end{array}}{\Delta \Vdash x = e.m(e_1, \ldots, e_n) : ((\kappa_s, \nu_s), (\kappa_h, \nu_h)) \parallel \mathsf{C}; \mathsf{D}}\ \text{[C-CALL]}$$

$$\frac{\Delta \Vdash x : (T_x\!<\!\overline{\rho_x}\!>, (\kappa_x, \nu_x)) \parallel \mathsf{C_x}; \mathsf{D_x} \quad (\eta, \mathcal{X}) = \mathit{glevel}(C\!<\!\overline{\rho_C}\!>) \quad \mathsf{C} = \mathsf{C_x} \cup \{\eta \sqsubseteq \kappa_x, \kappa_s \sqsubseteq \kappa_x, \kappa_h \sqsubseteq \eta\} \quad \mathsf{D} = \mathsf{D_x} \cup \{\mathcal{X} \subseteq \nu_x, \nu_s \subseteq \nu_x, \nu_h \subseteq \mathcal{X}\}}{\Delta \Vdash x = \text{new } C\!<\!\overline{\rho_C}\!>\!: ((\kappa_s, \nu_s), (\kappa_h, \nu_h)) \parallel \mathsf{C}; \mathsf{D}}\ \text{[C-NEW]}$$

$$\frac{\begin{array}{l}\Delta \Vdash e : (\text{Bool}, (\kappa_e, \nu_e)) \parallel \mathsf{C_e}; \mathsf{D_e} \\ \Delta \Vdash B_t : ((\kappa_s^t, \nu_s^t), (\kappa_h^t, \nu_h^t)) \parallel \mathsf{C_t}; \mathsf{D_t} \quad \Delta \Vdash B_f : ((\kappa_s^f, \nu_s^f), (\kappa_h^f, \nu_h^f)) \parallel \mathsf{C_f}; \mathsf{D_f} \\ \mathsf{C} = \mathsf{C_e} \cup \mathsf{C_t} \cup \mathsf{C_f} \cup \{\kappa_e \sqsubseteq \kappa_s, \kappa_s \sqsubseteq \kappa_s^t, \kappa_s \sqsubseteq \kappa_s^f, \kappa_e \sqsubseteq \kappa_h, \kappa_h \sqsubseteq \kappa_h^t, \kappa_h \sqsubseteq \kappa_h^f\} \\ \mathsf{D} = \mathsf{D_e} \cup \mathsf{D_t} \cup \mathsf{D_f} \cup \{\nu_e \subseteq \nu_s, \nu_s \subseteq \nu_s^t, \nu_s \subseteq \nu_s^f, \nu_e \subseteq \nu_h, \nu_h \subseteq \nu_h^t, \nu_h \subseteq \nu_h^f\}\end{array}}{\Delta \Vdash \text{if } (e)\ B_t \text{ else } B_f : ((\kappa_s, \nu_s), (\kappa_h, \nu_h)) \parallel \mathsf{C}; \mathsf{D}}\ \text{[C-IF]}$$

$$\frac{(T\!<\!\overline{\rho}\!>, (\eta, \mathcal{X})) = \Delta(x)}{\Delta \Vdash x : (T\!<\!\overline{\rho}\!>, (\eta, \mathcal{X})) \parallel \emptyset; \emptyset}\ \text{[C-VAR]} \qquad \frac{(T\!<\!\overline{\rho}\!>, (\eta, \mathcal{X})) = \Delta(\text{this})}{\Delta \Vdash \text{this} : (T\!<\!\overline{\rho}\!>, (\eta, \mathcal{X})) \parallel \emptyset; \emptyset}\ \text{[C-THIS]}$$

$$\frac{c \in \{\text{true}, \text{false}\}}{\Delta \Vdash c : (\text{Bool}, (\bot_{\mathcal{H}}, \emptyset)) \parallel \emptyset; \emptyset}\ \text{[C-BOOL]} \qquad \frac{}{\Delta \Vdash \text{null} : (\text{Null}, (\bot_{\mathcal{H}}, \emptyset)) \parallel \emptyset; \emptyset}\ \text{[C-NULL]}$$

$$\frac{\Delta \Vdash e : (T_e\!<\!\overline{\rho_e}\!>, (\kappa_e, \nu_e)) \parallel \mathsf{C_e}; \mathsf{D_e} \quad f : (T_f\!<\!\overline{\rho_f}\!>, (\eta, \mathcal{X})) \in \mathit{gsfields}(T_e\!<\!\overline{\rho_e}\!>) \quad \mathsf{C} = \mathsf{C_e} \cup \{\kappa_e \sqsubseteq \kappa, \eta \sqsubseteq \kappa\} \quad \mathsf{D} = \mathsf{D_e} \cup \{\nu_e \subseteq \nu, \mathcal{X} \subseteq \nu\}}{\Delta \Vdash e.f : (T_f\!<\!\overline{\rho_f}\!>, (\kappa, \nu)) \parallel \mathsf{C}; \mathsf{D}}\ \text{[C-FIELD]}$$

$$\frac{\Delta \Vdash e_1 : (T\!<\!\overline{\rho}\!>, (\kappa_1, \nu_1)) \parallel \mathsf{C_1}; \mathsf{D_1} \quad \Delta \Vdash e_2 : (T\!<\!\overline{\rho}\!>, (\kappa_2, \nu_2)) \parallel \mathsf{C_2}; \mathsf{D_2} \quad \mathsf{C} = \mathsf{C_1} \cup \mathsf{C_2} \cup \{\kappa_1 \sqsubseteq \kappa, \kappa_2 \sqsubseteq \kappa\} \quad \mathsf{D} = \mathsf{D_1} \cup \mathsf{D_2} \cup \{\nu_1 \subseteq \nu, \nu_2 \subseteq \nu\}}{\Delta \Vdash e_1\ ==\ e_2 : (\text{Bool}, (\kappa, \nu)) \parallel \mathsf{C}; \mathsf{D}}\ \text{[C-EQ]}$$

**図 3　制約集合の生成規則**

```
// (a)
@Target(ElementType.TYPE)
public @interface SecrecyVariables {
  String[] value();
}

// (b)
@Target({ElementType.TYPE,
  ElementType.TYPE_USE})
public @interface Secrecy {
  SecLattice value() default BOT;
  String[] params() default {};
}
```

```
// (c)
@Repeatable(Assigns.class)
@Target(ElementType.TYPE_USE)
public @interface Assign {
  String param();
  Secrecy arg();
}
@Target(ElementType.TYPE_USE)
public @interface Assigns {
  Assign[] value();
}
```

**図 4　Java アノテーション**

機密度定数に関する制約 $\kappa_1 \sqsubseteq \kappa_2$ と機密度パラメータの集合に関する制約 $\nu_1 \subseteq \nu_2$ の両辺が取り得る値は，それぞれ束 $(\mathcal{H}, \sqsubseteq)$ の要素と束 $(2^{\mathcal{Y}}, \subseteq)$ の要素であるため，図 3 のアルゴリズムが生成する 2 種類の制約集合はいずれも標準的な制約解消アルゴリズムによって充足可能性を判定できる．

## 4　Java アノテーション

情報流解析を行うためには変数の型などの一部として機密度を記述する必要があり，例えば $OO_G$ では専用の構文を備えている．しかし，既存のプログラミング言語に対して構文レベルの変更を行うことは容易ではなく，解決策の一つとして Java プログラムに対するアノテーションを活用した情報流解析の実現手法が提案されている [7]．本稿では，既存手法を拡張して機密度パラメータに対応した Java アノテーションを提案する．以下では紙数の制限により主要な変更点のみを示す．既存のアノテーションの全体像は [7] を参照されたい．既存手法に従い，機密度定数の束を表す列挙型は開発者が定義し，機密度を表すアノテーションなどこの列挙型に依存するアノテーションは自動生成する．

機密度パラメータの導入に伴い，クラス定義における機密度パラメータの宣言，機密度定数と機密度パラメータの集合の組としての機密度，インスタンス生成における機密度の機密度パラメータへの割り当て，の 3 点に対応する必要がある．これらに対応するアノテーションの定義を図 4 に示す．図 4(a) の@SecrecyVariables は機密度パラメータを宣言するためのアノテーションであり，クラス定義に付与される．図 4(b)(c) の 3 つのアノテーションは機密度定数の束を表す列挙型に（推移的に）依存するため自動生成の対象であり，ここでは列挙型を仮に SecLattice としている．(b) の@Secrecy は機密度を表し，要素 value が機密度定数，要素 params が機密度パラメータの集合を表す．デフォルト値の指定により機密度を $OO_G$ と同様な省略形で記述できる．BOT は最小の機密度定数を指す．(c) の@Assign は，インスタンス生成における機密度パラメータに対する機密度の割り当てを表す．要素 param が割り当てたい機密度パラメータ，要素 arg が割り当てる機密度を表す．@Assign は 1 つの機密度パラメータへの割り当てを表すため，クラスが複数の機密度パラメータを持つ場合は複数の@Assign を与える必要がある．Java の仕様では同一の構文要素に対して同名のアノテーションを複数与えられないため，この制限を回避するために標準で用意されている@Repeatable を利用する．デフォルトで指定されるため図 4 では省略されているが，これらのアノテーションの情報をクラスファイルに含めるため RetentionPolicy を CLASS とする．検査対象のプログラムが参照するクラスの機密度情報をコンパイルに必須のクラスファイルから取得するためである．

アノテーションの利用例と $OO_G$ プログラムとの対比を図 5 に示す．ここで, L と H が機密度定数である．Pair クラスの機密度パラメータ F と S を@SecrecyVariables

```
// Java
@SecrecyVariables({"F", "S"})
@Secrecy(L)
public class Pair {
  @Secrecy(params = "F") boolean first;
  @Secrecy(params = "S") boolean second;
}
```

```
// OOG
class Pair<F, S> L {
  (Bool, F) first;
  (Bool, S) second;
}
```

```
// Java
pair = new
       @Assign(param = "F", arg = @Secrecy(H))
       @Assign(param = "S", arg = @Secrecy(L))
       Pair();
```

```
// OOG
pair = new Pair<H, L>;
```

**図 5　Java アノテーションの利用例と $OO_G$ プログラムとの対応**

({"F", "S"}) のように文字列として宣言する．フィールド first の機密度が F であることを@Secrecy の params 要素に F を指定して記述する．省略せずに記述すれば@Secrecy(value = L, params = {"F"}) である．Pair クラスのインスタンス生成において，機密度パラメータ F に機密度 H を割り当てるために，@Assign を利用し param 要素を F，arg 要素を@Secrecy(H) とする．機密度パラメータ S に対する機密度 L の割り当ても同様である．

## 5　おわりに

本稿では，機密度パラメータに対応した情報流解析のための型検査アルゴリズムと Java アノテーションを提案した．型検査は制約集合の充足問題に帰着して実現される．機密度は機密度定数と機密度パラメータの集合の組であるため，機密度定数に関する制約集合と機密度パラメータの集合に関する制約集合を個別に生成し，ともに充足可能であれば型検査に成功と判定する．Java アノテーションは，情報流解析に必要な機密度を Java プログラム中に記述するための手段である．既存の情報流解析のためのアノテーションを拡張し，機密度を構成する機密度パラメータの集合の記述と，インスタンス生成における機密度の機密度パラメータへの割り当ての記述を可能にした．今後の課題として，制約付き機密度パラメータへの対応や型検査と Java アノテーションの実装が挙げられる．

**謝辞**　本研究の一部は JSPS 科研費 JP15K00112，JP17K12666，JP18K11241 および 2019 年度南山大学パッヘ研究奨励金 I-A-2 の助成による．

## 参考文献

[1] Anindya Banerjee and David A. Naumann. Secure Information Flow and Pointer Confinement in a Java-like Language. In *Proceedings of the 15th IEEE Computer Security Foundations Workshop*, pp. 253–267, 2002.

[2] 黒川翔, 桑原寛明, 山本晋一郎, 坂部俊樹, 酒井正彦, 草刈圭一朗, 西田直樹. 例外処理付きオブジェクト指向プログラムにおける情報流の安全性解析のための型システム. 電子情報通信学会論文誌 D, Vol. J91-D, No. 3, pp. 757–770, 2008.

[3] Andrei Sabelfeld and Andrew C. Myers. Language-Based Information-Flow Security. *IEEE Journal on Selected Areas in Communications*, Vol. 21, No. 1, pp. 5–19, 2003.

[4] Dennis Volpano, Geoffrey Smith, and Cynthia Irvine. A Sound Type System for Secure Flow Analysis. *Journal of Computer Security*, Vol. 4, No. 2, pp. 167–187, 1996.

[5] 吉田真也, 桑原寛明, 國枝義敏. オブジェクト指向言語の情報流解析における機密度のパラメータ化. コンピュータ ソフトウェア, Vol. 36, No. 1, pp. 48–65, 2019.

[6] 桑原寛明. 型検査に基づく手続き型言語向け情報流解析における型エラースライシング. コンピュータソフトウェア, Vol. 27, No. 4, pp. 221–227, 2010.

[7] 吉田真也, 桑原寛明, 國枝義敏. 情報流解析のための Java アノテーション. コンピュータ ソフトウェア, Vol. 34, No. 4, pp. 47–53, 2017.