Décision de l'équivalence observationnelle et système de contraintes

Vincent Cheval

Stage sous la direction de Hubert Comon-Lundh et Stéphanie Delaune

7 septembre 2009

Equivalence observationnelle et propriété de sécurité

Communiquer de manière sécurisée sur des canaux non sécurisés

Equivalence observationnelle et propriété de sécurité

Communiquer de manière sécurisée sur des canaux non sécurisés

Cryptographie + Protocole (= Programme distribué)

Equivalence observationnelle et propriété de sécurité

Communiquer de manière sécurisée sur des canaux non sécurisés

Cryptographie + Protocole (= Programme distribué)

Comment être sûr de l'absence d'attaque ?

Equivalence observationnelle et propriété de sécurité

Communiquer de manière sécurisée sur des canaux non sécurisés

Cryptographie + Protocole (= Programme distribué)

Comment être sûr de l'absence d'attaque ?

- Cryptographie fiable
- Spécification correcte
- Implémentation satisfaisant la spécification

Equivalence observationnelle et propriété de sécurité

Communiquer de manière sécurisée sur des canaux non sécurisés

Cryptographie + Protocole (= Programme distribué)

Comment être sûr de l'absence d'attaque ?

- Cryptographie fiable
- Spécification correcte
- Implémentation satisfaisant la spécification

Equivalence observationnelle et propriété de sécurité

Propriété de sécurité

- Propriété d'accessibilité : secret simple, authenticité,. . .
- Propriété d'équivalence observationnelle : secret fort, attaque par dictionnaire, anonymat, . . .
- . . .

Equivalence observationnelle et propriété de sécurité

Propriété de sécurité

- Propriété d'accessibilité : secret simple, authenticité,. . .
- Propriété d'équivalence observationnelle : secret fort, attaque par dictionnaire, anonymat, . . .
- . . .

Equivalence observationnelle et propriété de sécurité

Propriété de sécurité

- Propriété d'accessibilité : secret simple, authenticité,. . .
- Propriété d'équivalence observationnelle : secret fort, attaque par dictionnaire, anonymat, . . .
- . . .

Definition (Equivalence observationnelle)

$$P \approx Q \Leftrightarrow \forall R, P||R \text{ et } Q||R \text{ émettent les même signaux}$$

Equivalence observationnelle et propriété de sécurité : Exemple

Protocole Handshake

$$\begin{array}{lll} 0. & A \longrightarrow B : & enc(M, k_{ab}) \\ 1. & B \longrightarrow A : & enc(f(M), k_{ab}) \end{array}$$

Equivalence observationnelle et propriété de sécurité : Exemple

Protocole Handshake

$$\begin{array}{lll} 0. & A \longrightarrow B : & enc(M, k_{ab}) \\ 1. & B \longrightarrow A : & enc(f(M), k_{ab}) \end{array}$$

Propriété de sécurité : Attaque par dictionnaire Off-line

Après un nombre fini de sessions avec le serveur, l'attaquant tente de deviner la clé en essayant toutes les possibilités

Equivalence observationnelle et propriété de sécurité : Exemple

Protocole Handshake

$$\begin{array}{lll} 0. & A \longrightarrow B : & enc(M, k_{ab}) \\ 1. & B \longrightarrow A : & enc(f(M), k_{ab}) \end{array}$$

Propriété de sécurité : Attaque par dictionnaire Off-line

Après un nombre fini de sessions avec le serveur, l'attaquant tente de deviner la clé en essayant toutes les possibilités

Formellement

$$\nu k_{ab}.(P_A(k_{ab}) \| P_B(k_{ab}) \| P_A^2(k_{ab}) \| \ \ldots); c(k_{ab})$$
$$\approx$$
$$\nu k_{ab}.\nu k.(P_A(k_{ab}) \| P_B(k_{ab}) \| P_A^2(k_{ab}) \| \ \ldots); c(k)$$

Etat de l'art

Huttel (2002)

- Limitée au spi-calcul (primitives fixées)
- N'est pas implémentable (compléxité multi-exponentielle)
- Ne gère pas les propriétés de trace

Blanchet, Abadi, Fournet (2008)

- Nombre infini de sessions
- Diff-équivalence : Equivalence observationnelle entre deux processus ayant la même structure mais dont les messages sont différents
- Très efficace
- Algorithme pouvant ne pas terminer et donner de fausses attaques

Etat de l'art

Cortier, Delaune (2009) + Baudet (2005)

- Nombre borné de sessions
- Equivalence observationnelle entre deux processus positifs déterministes
- L'ensemble infini de traces est représenté par un système de contraintes
- Equivalence observationnelle $\Leftrightarrow$ équivalence symbolique de système de contraintes
- Algorithme de décision de l'équivalence symbolique de systèmes de contraintes positifs avec un théorie équationnelle sous-terme convergente

Objectif du stage

Objectif

- Rechercher un nouvel algorithme de décision de l'équivalence symbolique plus simple

  Réduction à l'équivalence de systèmes de contraintes en formes résolues
- Algorithme de décision de l'équivalence symbolique de systèmes de contraintes en formes résolues
- Implémentation

Objectif à long terme

- Extension à des systèmes de contraintes non positifs
- Extension à d'autres théories de l'intrus et d'autres primitives cryptographiques

Objectif du stage

Objectif

- Rechercher un nouvel algorithme de décision de l'équivalence symbolique plus simple

  Réduction à l'équivalence de systèmes de contraintes en formes résolues
- Algorithme de décision de l'équivalence symbolique de systèmes de contraintes en formes résolues
- Implémentation

Objectif à long terme

- Extension à des systèmes de contraintes non positifs
- Extension à d'autres théories de l'intrus et d'autres primitives cryptographiques

Objectif du stage

Objectif

- Rechercher un nouvel algorithme de décision de l'équivalence symbolique plus simple

  Réduction à l'équivalence de systèmes de contraintes en formes résolues
- Algorithme de décision de l'équivalence symbolique de systèmes de contraintes en formes résolues
- Implémentation

Objectif à long terme

- Extension à des systèmes de contraintes non positifs
- Extension à d'autres théories de l'intrus et d'autres primitives cryptographiques

Objectif du stage

Objectif

- Rechercher un nouvel algorithme de décision de l'équivalence symbolique plus simple

  Réduction à l'équivalence de systèmes de contraintes en formes résolues
- Algorithme de décision de l'équivalence symbolique de systèmes de contraintes en formes résolues
- Implémentation

Objectif à long terme

- Extension à des systèmes de contraintes non positifs
- Extension à d'autres théories de l'intrus et d'autres primitives cryptographiques

Systèmes de contraintes

Protocole Handshake

$$0. \quad A \longrightarrow B : \quad enc(M, k_{ab})$$
$$1. \quad B \longrightarrow A : \quad enc(f(M), k_{ab})$$

Systèmes de contraintes

Protocole Handshake

$$\begin{array}{lll} 0. & A \longrightarrow B : & enc(M, k_{ab}) \\ 1. & B \longrightarrow A : & enc(f(M), k_{ab}) \end{array}$$

Système de contrainte associé

$$enc(x, k_{ab})$$

Systèmes de contraintes

Protocole Handshake

$$\begin{array}{lll} 0. & A \longrightarrow B : & enc(M, k_{ab}) \\ 1. & B \longrightarrow A : & enc(f(M), k_{ab}) \end{array}$$

Système de contrainte associé

$$enc(M, k_{ab}) \quad \Vdash enc(x, k_{ab})$$

Systèmes de contraintes

Protocole Handshake

$$\begin{array}{lll} 0. & A \longrightarrow B : & enc(M, k_{ab}) \\ 1. & B \longrightarrow A : & enc(f(M), k_{ab}) \end{array}$$

Système de contrainte associé

$$\begin{array}{ll} enc(M, k_{ab}) & \Vdash enc(x, k_{ab}) \\ enc(M, k_{ab}), enc(f(x), k_{ab}) & \end{array}$$

Systèmes de contraintes

Protocole Handshake

$$\begin{array}{lll} 0. & A \longrightarrow B : & enc(M, k_{ab}) \\ 1. & B \longrightarrow A : & enc(f(M), k_{ab}) \end{array}$$

Système de contrainte associé

$$\begin{array}{ll} enc(M, k_{ab}) & \Vdash enc(x, k_{ab}) \\ enc(M, k_{ab}), enc(f(x), k_{ab}) & \Vdash enc(f(M), k_{ab}) \end{array}$$

Solution d'un système de contrainte

Système de contrainte associé

$$enc(M, k_{ab}) \qquad \Vdash enc(x, k_{ab})$$
$$enc(M, k_{ab}), enc(f(x), k_{ab}) \qquad \Vdash enc(f(M), k_{ab})$$

Solution

- $\sigma = \{x \to M\}$
- $\xi_1 = ax_1$
- $\xi_2 = ax_2$

Equivalence symbolique

Equivalence Statique : $S \sim S'$

Etant données deux séquences de termes S, S', l'attaquant ne peut pas les distinguer.

$$\forall (\xi, \xi') \in \Pi^2, \xi[S]\downarrow = \xi'[S]\downarrow \Leftrightarrow \xi[S']\downarrow = \xi'[S']\downarrow$$

$(S, C) \approx_s (S', C')$

Etant donnés deux systèmes de contraintes et deux séquences, les traces représentées sont en équivalence statique.

- $\forall (\sigma, \xi_1, \ldots, \xi_n) \in Sol(C), \exists \sigma' \text{ tq } (\sigma', \xi_1, \ldots, \xi_n) \in Sol(C') \wedge S\sigma \sim S'\sigma'$
- $\forall (\sigma', \xi_1, \ldots, \xi_n) \in Sol(C'), \exists \sigma \text{ tq } (\sigma, \xi_1, \ldots, \xi_n) \in Sol(C) \wedge S\sigma \sim S'\sigma'$

Exemple

Systèmes de contrainte (Attaque par dictionnaire)

$$\begin{aligned} enc(M, k_{ab}) &\Vdash enc(x, k_{ab}) \\ enc(M, k_{ab}), enc(f(x), k_{ab}) &\Vdash enc(f(M), k_{ab}) \end{aligned}$$

$$\begin{aligned} S &= enc(M, k_{ab}), enc(f(x), k_{ab}), k_{ab} \\ S' &= enc(M, k_{ab}), enc(f(x), k_{ab}), k \end{aligned}$$

Pas équivalent

- Une solution : $\sigma = \{x \to M\}, \xi_1 = ax_1, \xi_2 = ax_2$

Exemple

Systèmes de contrainte (Attaque par dictionnaire)

$$\begin{aligned} enc(M, k_{ab}) &\Vdash enc(x, k_{ab}) \\ enc(M, k_{ab}), enc(f(x), k_{ab}) &\Vdash enc(f(M), k_{ab}) \end{aligned}$$

$$\begin{aligned} S &= enc(M, k_{ab}), enc(f(x), k_{ab}), k_{ab} \\ S' &= enc(M, k_{ab}), enc(f(x), k_{ab}), k \end{aligned}$$

Pas équivalent

- Une solution : $\sigma = \{x \to M\}, \xi_1 = ax_1, \xi_2 = ax_2$
- $S\sigma \not\downarrow S'\sigma$: $\xi = f(dec(ax_1, ax_3)), \xi' = dec(ax_2, ax_3)$

Règles de réduction

Objectif de l'algorithme

Décider l'équivalence symbolique de systèmes de contraintes

Règles de réduction

Objectif de l'algorithme

Décider l'équivalence symbolique de systèmes de contraintes

Contribution

Ensemble de règles qui :

- transforme les systèmes de contraintes en systèmes plus simples (forme résolues)

Règles de réduction

Objectif de l'algorithme

Décider l'équivalence symbolique de systèmes de contraintes

Contribution

Ensemble de règles qui :

- transforme les systèmes de contraintes en systèmes plus simples (forme résolues)
- conserve l'équivalence symbolique des systèmes de contraintes

Règles de réduction

Objectif de l'algorithme

Décider l'équivalence symbolique de systèmes de contraintes

Contribution

Ensemble de règles qui :

- transforme les systèmes de contraintes en systèmes plus simples (forme résolues)
- conserve l'équivalence symbolique des systèmes de contraintes
- termine

Règles de réduction

$$R_1 : \{ T_i \Vdash f(t_1, t_2) \quad \nearrow \quad \begin{cases} T_i \Vdash t_1 \\ T_i \Vdash t_2 \end{cases}$$

$$\searrow \quad \{ T_i \Vdash f(t_1, t_2)$$

$$R_2 : \begin{cases} \mathcal{C}_0 \\ T_1, v, T_2 \Vdash u \\ \mathcal{C}_1 \end{cases} \quad \nearrow \quad \begin{cases} \mathcal{C}_0\alpha \\ \mathcal{C}_1\alpha \end{cases}$$

$$\alpha = mgu(u, v) \quad \searrow \quad \begin{cases} \mathcal{C}_0 \\ T_1, v, T_2 \Vdash u \\ \mathcal{C}_1 \end{cases}$$

$$R_3 : \left\{ T_0, \{v_1\}_{v_2}, T_2 \Vdash u_1 \right. \quad \nearrow \begin{cases} T_0, \{v_1\}_{v_2}, T_2 \Vdash v_2 \\ T_0, \{v_1\}_{v_2}, T_2, v_1 \Vdash u_1 \end{cases}$$

$$\searrow \left\{ T_0, \{v_1\}_{v_2}, T_2 \Vdash u_1 \right.$$

$$R_4 : \left\{ T_1, \langle v_1, v_2 \rangle, T_2 \Vdash u_1 \right. \longrightarrow \left\{ T_1, \langle v_1, v_2 \rangle, v_1, v_2, T_2 \Vdash u_1 \right.$$

Example

$$
\begin{array}{lll}
d, e & \Vdash & < x, y > \\
d, e & \Vdash & z \\
d, e, z & \Vdash & w \\
d, e, z, \{c\}_{<d,e>} & \Vdash & c \\
d, e, z, \{c\}_{<d,e>}, \{z\}_h & \Vdash & \{\{d\}_e\}_h \\
d, e, z, \{c\}_{<d,e>}, \{z\}_h, \{e\}_h & \Vdash & \{w\}_h
\end{array}
$$

$$
\begin{array}{lll}
a, b & \Vdash & < x, y > \\
a, b & \Vdash & z \\
a, b, \{a\}_b & \Vdash & w \\
a, b, \{a\}_b, \{c\}_{<a,b>} & \Vdash & c \\
a, b, \{a\}_b, \{c\}_{<a,b>}, \{z\}_f & \Vdash & \{\{a\}_b\}_f \\
a, b, \{a\}_b, \{c\}_{<a,b>}, \{z\}_f, \{b\}_f & \Vdash & \{w\}_f
\end{array}
$$

Example

$$
\begin{array}{lll}
d, e & \Vdash & x \\
d, e & \Vdash & y \\
d, e & \Vdash & z \\
d, e, z & \Vdash & w \\
d, e, z, \{c\}_{<d,e>} & \Vdash & c \\
d, e, z, \{c\}_{<d,e>}, \{z\}_h & \Vdash & \{\{d\}_e\}_h \\
d, e, z, \{c\}_{<d,e>}, \{z\}_h, \{e\}_h & \Vdash & \{w\}_h
\end{array}
$$

$$
\begin{array}{lll}
a, b & \Vdash & x \\
a, b & \Vdash & y \\
a, b & \Vdash & z \\
a, b, \{a\}_b & \Vdash & w \\
a, b, \{a\}_b, \{c\}_{<a,b>} & \Vdash & c \\
a, b, \{a\}_b, \{c\}_{<a,b>}, \{z\}_f & \Vdash & \{\{a\}_b\}_f \\
a, b, \{a\}_b, \{c\}_{<a,b>}, \{z\}_f, \{b\}_f & \Vdash & \{w\}_f
\end{array}
$$

Example

$$
\begin{array}{lcl}
d, e & \Vdash & z \\
d, e, z & \Vdash & w \\
d, e, z, \{c\}_{<d,e>} & \Vdash & < d, e > \\
d, e, z, \{c\}_{<d,e>}, c & \Vdash & c \\
d, e, z, \{c\}_{<d,e>}, c, \{z\}_h & \Vdash & \{\{d\}_e\}_h \\
d, e, z, \{c\}_{<d,e>}, c, \{z\}_h, \{e\}_h & \Vdash & \{w\}_h
\end{array}
$$

$$
\begin{array}{lcl}
a, b & \Vdash & z \\
a, b, \{a\}_b & \Vdash & w \\
a, b, \{a\}_b, \{c\}_{<a,b>} & \Vdash & < a, b > \\
a, b, \{a\}_b, \{c\}_{<a,b>}, c & \Vdash & c \\
a, b, \{a\}_b, \{c\}_{<a,b>}, c, \{z\}_f & \Vdash & \{\{a\}_b\}_f \\
a, b, \{a\}_b, \{c\}_{<a,b>}, c, \{z\}_f, \{b\}_f & \Vdash & \{w\}_f
\end{array}
$$

Example

$$
\begin{array}{lcl}
d, e & \Vdash & z \\
d, e, z & \Vdash & w \\
d, e, z, \{c\}_{<d,e>} & \Vdash & <d, e> \\
d, e, z, \{c\}_{<d,e>}, c & \Vdash & c \\
d, e, z, \{c\}_{<d,e>}, c, \{z\}_h & \Vdash & \{\{d\}_e\}_h \\
d, e, z, \{c\}_{<d,e>}, c, \{z\}_h, \{e\}_h & \Vdash & \{w\}_h
\end{array}
$$

$$
\begin{array}{lcl}
a, b & \Vdash & \{a\}_b \\
a, b, \{a\}_b & \Vdash & w \\
a, b, \{a\}_b, \{c\}_{<a,b>} & \Vdash & <a, b> \\
a, b, \{a\}_b, \{c\}_{<a,b>}, c & \Vdash & c \\
a, b, \{a\}_b, \{c\}_{<a,b>}, c, \{\{a\}_b\}_f & \Vdash & \{\{a\}_b\}_f \\
a, b, \{a\}_b, \{c\}_{<a,b>}, c, \{\{a\}_b\}_f, \{b\}_f & \Vdash & \{w\}_f
\end{array}
$$

Example

$$
\begin{array}{lcl}
d, e & \Vdash & \{d\}_e \\
d, e, \{d\}_e & \Vdash & w \\
d, e, \{d\}_e, \{c\}_{<d,e>} & \Vdash & <d, e> \\
d, e, \{d\}_e, \{c\}_{<d,e>}, c & \Vdash & c \\
d, e, \{d\}_e, \{c\}_{<d,e>}, c, \{\{d\}_e\}_h & \Vdash & \{\{d\}_e\}_h \\
d, e, \{d\}_e, \{c\}_{<d,e>}, c, \{\{d\}_e\}_h, \{e\}_h & \Vdash & \{w\}_h
\end{array}
$$

$$
\begin{array}{lcl}
a, b & \Vdash & \{a\}_b \\
a, b, \{a\}_b & \Vdash & w \\
a, b, \{a\}_b, \{c\}_{<a,b>} & \Vdash & <a, b> \\
a, b, \{a\}_b, \{c\}_{<a,b>}, c & \Vdash & c \\
a, b, \{a\}_b, \{c\}_{<a,b>}, c, \{\{a\}_b\}_f & \Vdash & \{\{a\}_b\}_f \\
a, b, \{a\}_b, \{c\}_{<a,b>}, c, \{\{a\}_b\}_f, \{b\}_f & \Vdash & \{w\}_f
\end{array}
$$

Example

$$
\begin{array}{lll}
d, e & \Vdash & \{d\}_e \\
d, e, \{d\}_e & \Vdash & e \\
d, e, \{d\}_e, \{c\}_{<d,e>} & \Vdash & <d, e> \\
d, e, \{d\}_e, \{c\}_{<d,e>}, c & \Vdash & c \\
d, e, \{d\}_e, \{c\}_{<d,e>}, c, \{\{d\}_e\}_h & \Vdash & \{\{d\}_e\}_h \\
d, e, \{d\}_e, \{c\}_{<d,e>}, c, \{\{d\}_e\}_h, \{e\}_h & \Vdash & \{e\}_h
\end{array}
$$

$$
\begin{array}{lll}
a, b & \Vdash & \{a\}_b \\
a, b, \{a\}_b & \Vdash & b \\
a, b, \{a\}_b, \{c\}_{<a,b>} & \Vdash & <a, b> \\
a, b, \{a\}_b, \{c\}_{<a,b>}, c & \Vdash & c \\
a, b, \{a\}_b, \{c\}_{<a,b>}, c, \{\{a\}_b\}_f & \Vdash & \{\{a\}_b\}_f \\
a, b, \{a\}_b, \{c\}_{<a,b>}, c, \{\{a\}_b\}_f, \{b\}_f & \Vdash & \{b\}_f
\end{array}
$$

Example

$$
\begin{array}{lcl}
d, e & \Vdash & \{d\}_e \\
d, e, \{d\}_e & \Vdash & w \\
d, e, \{d\}_e, \{c\}_{<d,e>} & \Vdash & <d, e> \\
d, e, \{d\}_e, \{c\}_{<d,e>}, c & \Vdash & c \\
d, e, \{d\}_e, \{c\}_{<d,e>}, c, \{\{d\}_e\}_h & \Vdash & \{\{d\}_e\}_h \\
d, e, \{d\}_e, \{c\}_{<d,e>}, c, \{\{d\}_e\}_h, \{e\}_h & \Vdash & \{w\}_h
\end{array}
$$

$$
\begin{array}{lcl}
a, b & \Vdash & \{a\}_b \\
a, b, \{a\}_b & \Vdash & w \\
a, b, \{a\}_b, \{c\}_{<a,b>} & \Vdash & <a, b> \\
a, b, \{a\}_b, \{c\}_{<a,b>}, c & \Vdash & c \\
a, b, \{a\}_b, \{c\}_{<a,b>}, c, \{\{a\}_b\}_f & \Vdash & \{\{a\}_b\}_f \\
a, b, \{a\}_b, \{c\}_{<a,b>}, c, \{\{a\}_b\}_f, \{b\}_f & \Vdash & \{w\}_f
\end{array}
$$

Example

$$
\begin{array}{lcl}
d, e & \Vdash & \{d\}_e \\
d, e, \{d\}_e & \Vdash & \{d\}_e \\
d, e, \{d\}_e, \{c\}_{<d,e>} & \Vdash & <d, e> \\
d, e, \{d\}_e, \{c\}_{<d,e>}, c & \Vdash & c \\
d, e, \{d\}_e, \{c\}_{<d,e>}, c, \{\{d\}_e\}_h & \Vdash & \{\{d\}_e\}_h \\
d, e, \{d\}_e, \{c\}_{<d,e>}, c, \{\{d\}_e\}_h, \{e\}_h & \Vdash & \{\{d\}_e\}_h
\end{array}
$$

$$
\begin{array}{lcl}
a, b & \Vdash & \{a\}_b \\
a, b, \{a\}_b & \Vdash & \{a\}_b \\
a, b, \{a\}_b, \{c\}_{<a,b>} & \Vdash & <a, b> \\
a, b, \{a\}_b, \{c\}_{<a,b>}, c & \Vdash & c \\
a, b, \{a\}_b, \{c\}_{<a,b>}, c, \{\{a\}_b\}_f & \Vdash & \{\{a\}_b\}_f \\
a, b, \{a\}_b, \{c\}_{<a,b>}, c, \{\{a\}_b\}_f, \{b\}_f & \Vdash & \{\{a\}_b\}_f
\end{array}
$$