Trace Equivalence and Epistemic Logic to Express Security Properties

Kiraku Minami(✉)

Kyoto University, Kyoto 606-8502, Japan
kminami@kurims.kyoto-u.ac.jp

Abstract. In process algebra, we can express security properties using an equivalence on processes. However, it is not clear which equivalence is the most suitable for the purpose. Indeed, several definitions of some properties are proposed. For example, the definition of privacy is not unique. This situation means that we are not certain how to express an intuitive security notion. Namely, there is a gap between an intuitive security notion and the formulation. Proper formalization is essential for verification, and our purpose is to bridge this gap.

In the case of the applied pi calculus, an outputted message is not explicitly expressed. This feature suggests that trace equivalence appropriately expresses indistinguishability for attackers in the applied pi calculus. By chasing interchanging bound names and scope extrusions, we prove that trace equivalence is a congruence. Therefore, a security property expressed using trace equivalence is preserved by application of contexts.

Moreover, we construct an epistemic logic for the applied pi calculus. We show that its logical equivalence agrees with trace equivalence. It means that trace equivalence is suitable in the presence of a non-adaptive attacker. Besides, we define several security properties to use our epistemic logic.

Keywords: Applied pi calculus · Trace equivalence · Epistemic logic

1 Introduction

1.1 Background

In modern society, information technology is indispensable to our daily lives, and many communication protocols are developed to transmit data securely. Verification of security properties of each protocol is essential, but it is not easy.

In the first place, how to formalize security notions is not clear. Various definitions of the same security property have been proposed, we will show an example later. One of our goals is to provide foundations to express these properties in

This work was partly supported by JST ERATO Grant Number JPMJER1603, Japan.

a rigorous way. Besides, how to model communication and concurrency is also not clear; many such models have also been developed. In this work, we focus on process algebra because it allows us to handle parallel composition naturally.

In process calculi, common confidentiality properties such as secrecy are represented to use an equivalence on processes. Many equivalences exist (cf. [18]), but which is the most suitable for expressing confidentiality is not clear. For instance, Delaune et al. [13] defined privacy in electronic voting in terms of the applied pi calculus [1] as follows.

Definition 1 ([13, Definition 9]). *A voting protocol respects vote-privacy (or just privacy) if*

$$S[V_A\{{}^{a}/_{v}\}|V_B\{{}^{b}/_{v}\}] \approx_l S[V_A\{{}^{b}/_{v}\}|V_B\{{}^{a}/_{v}\}]$$

for all possible votes a and b.

V_A and V_B denote the voters containing the free variable v. S is an evaluation context. S denotes other voters and authorities. Intuitively, when the protocol respects privacy, this definition states that an attacker cannot distinguish two situations where votes are swapped. Note that indistinguishability is expressed to use labeled bisimilarity $\approx_l$ in this definition. Is it the most suitable? This question is nontrivial. Indeed, Chadha et al. [7] gave another definition and claimed that trace equivalence is more suitable than bisimilarity to model privacy. We also claim that trace equivalence is more suitable to express security properties in the presence of non-adaptive attackers. Similar arguments are not abundant in previous work.

In the applied pi calculus, a process can send not only names but also terms, but we do not explicitly express sent messages. We indirectly represent them to use alias variables. This feature enables us to handle cryptographic protocols naturally and suggests that trace equivalence means indistinguishability for attackers. This is because attackers whom we consider can observe only labeled transitions. We recall the syntax and semantics of the applied pi calculus in Sect. 2.

Both bisimilarity and trace equivalence on labeled transition systems (LTSs) are well studied. However, trace equivalence in the applied pi calculus (and other variants of the pi calculus [26,27]) has not drawn much attention. This is perhaps because trace equivalence is the coarsest among commonly used equivalences. However, security properties sometimes require that different processes are regarded as the same. For example, consider secrecy. We want to make two processes that send different messages indistinguishable. In the case, trace equivalence is enough, but bisimilarity is not always optimal because it is too strong. Bisimilarity requires that possible actions for each process are the same. However, a non-adaptive attacker cannot detect a difference in feasibility. Here, "non-adaptive" means that the attacker cannot control participants. Thereby, a fine equivalence such as bisimilarity is not always adequate. Bisimilarity is probably suitable for more powerful attackers.

Epistemic logic is often used to express confidentiality directly (e.g. [7,25,32]). For example, when a message M sent by an agent a is anonymous, we might say that an adversary cannot know who sent M. In epistemic logic, we can express it with a formula such as $\neg K\text{Send}(a, M)$. This logical formulation is close to our intuition. Nevertheless, research into an epistemic logic for the applied pi calculus is not abundant.

In this paper, we assume that attackers can observe only labeled transitions. Especially, they cannot observe what action participants' can do. This assumption is natural because attackers in this paper are non-adaptive. We also assume that an attacker can send messages to participants.

1.2 Contributions

We prove that trace equivalence for the applied pi calculus is a congruence in Sect. 3. Second, we give an epistemic logic that characterizes trace equivalence in Sect. 4. Besides, we define security properties such as role-interchangeability, secrecy [25,32], and openness, using our epistemic logic. Moreover, we show that parallel composition does not generally preserve secrecy and openness.

Whereas, trace equivalence characterizes total secrecy, so application of contexts preserves it. We omit many proofs. See [28] for details.

Our results suggest that trace equivalence is more suitable to express (non-probabilistic) security notions than bisimilarity.

2 The Applied Pi Calculus

The applied pi calculus [1] is an extension of the pi-calculus [26,27]. We can handle cryptographic protocols naturally to use it.

2.1 Syntax

Let Σ be a signature equipped with an equational theory. Terms are made from names, variables, and function symbols. A term is ground when it contains no variables. We recall the syntax of the applied pi-calculus. Here, M, N... range over terms, while n on names and x on variables.

$$\begin{aligned} P, Q ::= \; & 0 \mid \overline{M}\langle N\rangle.P \mid M(x).P \mid \nu n.P \mid \\ & \qquad \text{if } M = N \text{ then } P \text{ else } Q \mid P + Q \mid P|Q \mid !P \end{aligned}$$

$$A, B ::= P \mid \nu n.A \mid \nu x.A \mid A|B \mid \{^M/_x\}$$

P, Q, ... are plain processes. ν is a binding operator. $|$ is a parallel composition operator. $+$ is a nondeterministic choice operator. Plain processes are similar to pi-calculus processes, but they are not the same. A pi-calculus process can send only a name. On the other hand, an applied pi calculus process can send a term. Besides, a channel consists of a term. An object of an input prefix is a variable, so names do not change while the process runs.

$A, B, ...$ are extended processes. We call $\{{}^{M}/_{x}\}$ an active substitution. This notion is peculiar to the applied pi calculus. An active substitution $\{{}^{M}/_{x}\}$ substitutes M for x in a neighbor process.

fn(A) and bn(A) denote the sets of free names and bound names of a process A, respectively. fv and bv are similar to them. If fn$(A) \cap$ bn$(A) = \emptyset$ and no bound names are restricted twice, we say that A is name-distinct. Variable-distinctness is defined similarly. n(M) denotes the set of names that appear in a term M. v(M) is similar to it.

The domain dom(A) of an extended process A is inductively defined below. If variables in neighbor concurrently running processes are in dom(A), the process A affects those variables. If fv$(A) =$ dom(A), we say that A is closed.

$$\mathrm{dom}(P) = \emptyset,\ \mathrm{dom}(\nu n.A) = \mathrm{dom}(A),\ \mathrm{dom}(\nu x.A) = \mathrm{dom}(A) \setminus \{x\},$$
$$\mathrm{dom}(A|B) = \mathrm{dom}(A) \cup \mathrm{dom}(B),\ \mathrm{dom}(\{{}^{M}/_{x}\}) = \{x\}$$

2.2 Semantics

A context is an expression containing one hole. An evaluation context is a context whose hole is neither under a replication, a conditional branch, nor an action prefix. Structural equivalence $\equiv$ is the smallest equivalence relation on extended processes that is closed under application of evaluation contexts and α-conversion, such that:

$$A|0 \equiv A \quad (A|B)|C \equiv A|(B|C) \quad A|B \equiv B|A$$
$$(\nu u.A)|B \equiv \nu u.(A|B) \ \text{ if } u \notin \mathrm{fn}(B) \cup \mathrm{fv}(B) \quad \nu u.\nu v.A \equiv \nu v.\nu u.A \quad !P \equiv P|!P$$
$$P + Q \equiv Q + P \quad \nu x.\{{}^{M}/_{x}\} \equiv 0 \quad A|\{{}^{M}/_{x}\} \equiv A[{}^{M}/_{x}]|\{{}^{M}/_{x}\}$$
$$\{{}^{M}/_{x}\} \equiv \{{}^{N}/_{x}\} \ \text{ if } \Sigma \vdash M = N;$$

The second from the last represents how an active substitution $\{{}^{M}/_{x}\}$ acts.

Definition 2. *Internal reduction $\rightarrow$ is the smallest relation on extended processes closed under structural equivalence and application of evaluation contexts, such that:*

$$\begin{array}{rl} \text{if } M = N \text{ then } P \text{ else } Q \rightarrow P & \text{when } \Sigma \vdash M = N \\ \text{if } M = N \text{ then } P \text{ else } Q \rightarrow Q & \text{when } \Sigma \nvdash M = N \\ P + Q \rightarrow P & \\ \overline{M}\langle N\rangle.P \mid M(x).Q \rightarrow P \mid Q[{}^{N}/_{x}], & \end{array}$$

where terms M and N in the second rule are ground.

The last line represents synchronous communication on a channel M. We emphasize that an environment cannot observe what is interchanged.

Next, we recall labeled semantics and requisite equivalence relations.

$$\frac{}{M(x).P \xrightarrow{M(N)} P[^N/_x]} \qquad \frac{x \notin \mathrm{fv}(\overline{M}\langle N\rangle.P)}{\overline{M}\langle N\rangle.P \xrightarrow{\nu x.\overline{M}\langle x\rangle} P|\{^N/_x\}}$$

$$\frac{A \xrightarrow{\alpha} A' \quad u \text{ does not appear in } \alpha.}{\nu u.A \xrightarrow{\alpha} \nu u.A'}$$

$$\frac{A \xrightarrow{\alpha} A' \quad \mathrm{bv}(\alpha) \cap \mathrm{fv}(B) = \emptyset}{A|B \xrightarrow{\alpha} A'|B} \qquad \frac{A \equiv A' \quad A' \xrightarrow{\alpha} B' \quad B' \equiv B}{A \xrightarrow{\alpha} B}$$

The second rule represents an output. Note that an active substitution $\{^N/_x\}$ is generated. The term N does not appear in the action label $\nu x.\overline{M}\langle x\rangle$.

A frame is an extended process generated from 0 and active substitutions to use restriction and parallel composition. fr(A) denotes a process obtained by replacing plain processes in A with 0, and we call it a frame of A. We can consider that fr(A) is a list of outputted messages.

μ is an action. We define $\Longrightarrow$ as the transitive reflexive closure of $\longrightarrow$, and $\overset{\alpha}{\Longrightarrow}$ as $\Longrightarrow\xrightarrow{\alpha}\Longrightarrow$. $\overset{\mu}{\Longrightarrow}$ is the former when μ is silent and is the latter otherwise.

Definition 3. *$(M = N)\varphi \overset{\mathrm{def}}{\Leftrightarrow} \mathrm{v}(M) \cup \mathrm{v}(N) \subseteq \mathrm{dom}(\varphi)$ and $M\sigma = N\sigma$ where $\varphi \equiv \nu\widetilde{n}.\sigma$ and $\widetilde{n} \cap \mathrm{n}(M, N) = \emptyset$ for some names $\widetilde{n}$ and active substitutions σ.*

$(M = N)\varphi$ means that an attacker cannot distinguish M and N to use φ.

Definition 4. *The static equivalence on closed frames is given by*

$$\varphi \approx_s \psi \overset{\mathrm{def}}{\Leftrightarrow} \mathrm{dom}(\varphi) = \mathrm{dom}(\psi) \text{ and } \forall M, N; (M = N)\varphi \Leftrightarrow (M = N)\psi$$

for closed frames φ and ψ. The static equivalence on closed processes is given by

$$A \approx_s B \overset{\mathrm{def}}{\Leftrightarrow} \mathrm{fr}(A) \approx_s \mathrm{fr}(B)$$

for closed processes A and B.

Static equivalence means that an attacker has the same information about which terms are equal.

Definition 5. *A trace* $\mathbf{tr}$ *is a finite derivation $\mathbf{tr} = A_0 \overset{\mu_1}{\Longrightarrow} \dots \overset{\mu_n}{\Longrightarrow} A_n$ such that every A_i is closed and $\mathrm{fv}(\mu_i) \subseteq \mathrm{dom}(A_{i-1})$ for all i. If A_n can perform no actions, the trace* $\mathbf{tr}$ *is said to be complete or maximal. Given a trace* $\mathbf{tr}$*, let $\mathbf{tr}[i]$ be its i-th process A_i, and $\mathbf{tr}[i, j]$ be the trace $A_i \overset{\mu_{i+1}}{\Longrightarrow} \dots \overset{\mu_j}{\Longrightarrow} A_j$ where $0 \le i \le j \le n$. The length of the trace* $\mathbf{tr}$ *is denoted by $|\mathbf{tr}| = n$.*

Definition 6. *Let* $\mathbf{tr}$ *be a trace $A_0 \overset{\mu_1}{\Longrightarrow} \dots \overset{\mu_n}{\Longrightarrow} A_n$ and $\mathbf{tr}'$ be a trace $B_0 \overset{\mu'_1}{\Longrightarrow} \dots \overset{\mu'_m}{\Longrightarrow} B_m$. Static equivalence between* $\mathbf{tr}$ *and* $\mathbf{tr}'$ *is defined as below:*

$\mathbf{tr} \sim_t \mathbf{tr}' \overset{\mathrm{def}}{\Leftrightarrow} n = m$ *and* $\mu_i = \mu'_i$ *and* $A_i \approx_s B_i$ *for all* i.

An attacker cannot distinguish statically equivalent traces. tr(A) is a set of traces of A. $\mathrm{tr}_{\max}(A)$ is a set of maximal traces of A.

Definition 7. *Let A and B be closed processes.*

$$A \subseteq_t B \overset{\mathrm{def}}{\Leftrightarrow} \forall \mathbf{tr} \in \mathrm{tr}(A)\ \exists \mathbf{tr'} \in \mathrm{tr}(B) \text{ s.t. } \mathbf{tr} \sim_t \mathbf{tr'},$$
$$A \approx_t B \overset{\mathrm{def}}{\Leftrightarrow} A \subseteq_t B \textit{ and } B \subseteq_t A.$$

Let A and B be two processes. Let σ be a map that maps variables in $(\mathrm{fv}(A) \setminus \mathrm{dom}(A)) \cup (\mathrm{fv}(B) \setminus \mathrm{dom}(B))$ to ground terms. When $A\sigma \approx_t B\sigma$ holds for every such σ, we also denote it as $A \approx_t B$.

$A \subseteq_t B$ means that each trace of A is imitated by some trace of B.

We later show that non-adaptive active attackers cannot distinguish trace equivalent processes.

Trace equivalence is undecidable. However, if processes contain no replications and the equational theory on Σ is a subterm convergent destructor rewriting system, trace equivalence is coNEXP complete [9].

3 Congruency of Trace Equivalence

The theorem below is our main result. It holds that trace equivalence is a congruence even though trace equivalence for the pi-calculus is not a congruence. This is ascribed to the difference between the pi-calculus and the applied pi calculus, namely, to the fact that names and variables are distinguished in the applied pi calculus. This is why adding an input prefix does not break trace equivalence. Besides, a scheme of late instantiation for an input transition is used in pi-calculus [26,27], so parallel composition may break trace equivalence. On the other hand, a scheme of early instantiation is used in the applied pi calculus. This scheme enables us to decompose a trace of a parallel composed process into traces of component processes.

Example 1. We consider pi-calculus. We put

$$P = z(z')|\overline{y}y'.\overline{w}w',\ Q = z(z').\overline{y}y'.\overline{w}w' + \overline{y}y'.z(z').\overline{w}w' + \overline{y}y'.\overline{w}w'.z(z').$$

Then, $x(z).P$ and $x(z).Q$ are trace equivalent, but $\overline{x}y|x(z).P$ and $\overline{x}y|x(z).Q$ are not trace equivalent. On the other hand, $x(z).P$ and $x(z).Q$ are not trace equivalent in the applied pi calculus because instantiation is early.

Abadi et al. [1] defined partial normal forms to prove that labeled bisimilarity is closed by application of closing evaluation contexts. They gave an operational semantics on partial normal forms. They classified transitions between ordinal processes into six cases to use partial normal forms.

To prove the next theorem, we use partial normal forms and define concurrent normal forms of traces. Transitions in a concurrent normal trace have to be particular forms.

Abadi et al. [1] studied decomposition and composition of reductions on partial normal forms. We study decomposition and composition of concurrent normal traces.

Theorem 1. $\approx_t$ *is a congruence.*

The proof is very long and complicated, so we only present an outline of our proof for the proposition below. Other cases are easy. The proof is given in [28].

Proposition 1. $A \approx_t B \Rightarrow A|C \approx_t B|C$.

Proof Outline. First, we define concurrent normal forms. A concurrent normal form is a particular form of a trace of a parallel composed process. A concurrent normal trace captures changes of scopes of bound names. Each process in a concurrent normal trace is of the form $\nu\widetilde{rs}.(\nu\widetilde{x}.(\sigma|P)\rho \mid \nu\widetilde{y}.(\rho|Q)\sigma)$, where σ and ρ are (active) substitutions. Terms sent by the left process are accumulated in σ. Bound names sent by the left process P are accumulated in $\widetilde{s}$. Symmetric cases are similar.

Second, for any trace t of $A|C$, we prove that there exists a concurrent normal trace t' of $A|C$ such that $t \sim_t t'$.

Third, given a concurrent normal trace **tr** of $A|C$, we prove that we can construct traces of A and C which each process in them is of the form $\nu\widetilde{s}.(\sigma|P)\rho$ or $\nu\widetilde{r}.(\rho|Q)\sigma$.

Finally, we take a trace **tr**$'$ of B which is statically equivalent to the extracted trace of A as the above, combine it with **tr**$'$, and prove that the result is statically equivalent to the given trace **tr**. □

Example 2. Let h be a unary function symbol which cannot be inverted. $\nu m.a(x).\overline{x}\langle m\rangle \approx_t \nu m.a(x).\overline{x}\langle h(m)\rangle$ holds. Then,

$$\nu m.a(x).\overline{x}\langle m\rangle \mid \nu n.\overline{a}\langle n\rangle.n(y).\overline{b}\langle y\rangle \approx_t \nu m.a(x).\overline{x}\langle h(m)\rangle \mid \nu n.\overline{a}\langle n\rangle.n(y).\overline{b}\langle y\rangle$$

is shown as follows.

We arbitrarily take a trace **tr** of the left hand side. We consider

$$\begin{aligned}
\mathbf{tr}\ :&\nu m.a(x).\overline{x}\langle m\rangle \mid \nu n.\overline{a}\langle n\rangle.n(y).\overline{b}\langle y\rangle\\
\xrightarrow{\nu z.\overline{a}\langle z\rangle}&\nu m.a(x).\overline{x}\langle m\rangle \mid \nu n.(n(y).\overline{b}\langle y\rangle \mid \{^n/_z\})\\
\xrightarrow{a(z)}&\nu mn.(\overline{n}\langle m\rangle \mid n(y).\overline{b}\langle y\rangle \mid \{^n/_z\})\\
\longrightarrow&\nu mn.(\overline{b}\langle m\rangle \mid \{^n/_z\})\\
\xrightarrow{\nu w.\overline{b}\langle w\rangle}&\nu mn.\{^n/_z, {}^m/_w\}
\end{aligned}$$

as an example. We transform it into a concurrent normal form.

$$\begin{aligned}
\mathbf{tr}'\ :&\nu m.a(x).\overline{x}\langle m\rangle \mid \nu n.\overline{a}\langle n\rangle.n(y).\overline{b}\langle y\rangle\\
\xrightarrow{\nu z.\overline{a}\langle z\rangle}&\nu n.((\nu m.a(x).\overline{x}\langle m\rangle)[^n/_z] \mid n(y).\overline{b}\langle y\rangle \mid \{^n/_z\})\\
\xrightarrow{a(z)}&\nu n.((\nu m.\overline{z}\langle m\rangle)[^n/_z] \mid n(y).\overline{b}\langle y\rangle \mid \{^n/_z\})\\
\longrightarrow&\nu nm.((\nu v.\{^m/_v\})[^n/_z] \mid (\overline{b}\langle v\rangle \mid \{^n/_z\})[^m/_v])\\
\xrightarrow{\nu w.\overline{b}\langle w\rangle}&\nu nm.((\nu v.\{^m/_v\})[^n/_z, {}^v/_w] \mid \{^n/_z, {}^v/_w\}[^m/_v])
\end{aligned}$$

Next, we decompose it into traces of component processes.

$$\begin{aligned}\mathbf{tr_1} :&\nu m.a(x).\overline{x}\langle m\rangle \\ \xrightarrow{a(n)}&(\nu m.\overline{z}\langle m\rangle)[{}^{n}/_{z}] \\ \xrightarrow{\nu v.\overline{n}\langle v\rangle}&(\nu m.\{{}^{m}/_{v}\})[{}^{n}/_{z}]\end{aligned}
\qquad
\begin{aligned}\mathbf{tr_2} :&\nu n.\overline{a}\langle n\rangle.n(y).\overline{b}\langle y\rangle \\ \xrightarrow{\nu z.\overline{a}\langle z\rangle}&\nu n.(n(y).\overline{b}\langle y\rangle \mid \{{}^{n}/_{z}\}) \\ \xrightarrow{z(m)}&\nu n.(\overline{b}\langle v\rangle \mid \{{}^{n}/_{z}\})[{}^{m}/_{v}] \\ \xrightarrow{\nu w.\overline{b}\langle w\rangle}&\nu n.\{{}^{n}/_{z}, {}^{v}/_{w}\}[{}^{m}/_{v}]\end{aligned}$$

Since $\nu m.a(x).\overline{x}\langle m\rangle \approx_t \nu m.a(x).\overline{x}\langle h(m)\rangle$ holds, we can take a trace of $\nu m.a(x).\overline{x}\langle h(m)\rangle$ which is statically equivalent to the former.

$$\begin{aligned}\mathbf{tr_3} :&\nu m.a(x).\overline{x}\langle h(m)\rangle \\ \xrightarrow{a(n)}&(\nu m.\overline{z}\langle h(m)\rangle)[{}^{n}/_{z}] \\ \xrightarrow{\nu v.\overline{n}\langle v\rangle}&(\nu m.\{{}^{h(m)}/_{v}\})[{}^{n}/_{z}]\end{aligned}$$

Finally, we compose $\mathbf{tr_2}$ and $\mathbf{tr_3}$ and obtain a desired trace $\mathbf{tr}_4$.

$$\begin{aligned}\mathbf{tr_4} :&\nu m.a(x).\overline{x}\langle h(m)\rangle \mid \nu n.\overline{a}\langle n\rangle.n(y).\overline{b}\langle y\rangle \\ \xrightarrow{\nu z.\overline{a}\langle z\rangle}&\nu n.((\nu m.a(x).\overline{x}\langle h(m)\rangle)[{}^{n}/_{z}] \mid n(y).\overline{b}\langle y\rangle \mid \{{}^{n}/_{z}\}) \\ \xrightarrow{a(z)}&\nu n.((\nu m.\overline{z}\langle h(m)\rangle)[{}^{n}/_{z}] \mid n(y).\overline{b}\langle y\rangle \mid \{{}^{n}/_{z}\}) \\ \longrightarrow&\nu nm.((\nu v.\{{}^{h(m)}/_{v}\})[{}^{n}/_{z}] \mid (\overline{b}\langle v\rangle \mid \{{}^{n}/_{z}\})[{}^{h(m)}/_{v}]) \\ \xrightarrow{\nu w.\overline{b}\langle w\rangle}&\nu nm.((\nu v.\{{}^{h(m)}/_{v}\})[{}^{n}/_{z}, {}^{v}/_{w}] \mid \{{}^{n}/_{z}, {}^{v}/_{w}\}[{}^{h(m)}/_{v}])\end{aligned}$$

Cheval et al. [10] established congruence property of trace equivalence for image-finite processes. They proved that trace equivalence is equivalent to may-testing equivalence for image-finite processes. On the other hand, taking all processes into account, may-testing equivalence does not imply trace equivalence. They gave a concrete counterexample. Thus, we cannot use the same technique.

4 An Epistemic Logic for the Applied Pi Calculus

4.1 Syntax

We propose an epistemic logic for the applied pi calculus. It was inspired by [7], but our logic is a bit different. We give syntax of formulas.

$$\begin{aligned}\delta ::=&\top | M_1 = M_2 | M \in \mathrm{dom} | \delta_1 \vee \delta_2 | \neg\delta \\ \varphi ::=&\delta | \varphi_1 \vee \varphi_2 | \neg\varphi | \langle\mu\rangle_- \varphi | F\varphi | K\varphi\end{aligned}$$

where M_1, M_2 and M are terms, and μ is an action. We call δ a static formula and φ a modal formula. A static formula δ mentions equality of terms. A modal formula φ mentions traces.

$\langle\mu\rangle_-\varphi$ states that the previous action is μ, and φ holds just before observing μ. $F\varphi$ states that φ holds some time or other. The operator K expresses an attacker's knowledge, i.e., $K\varphi$ means an attacker knows that φ holds.

4.2 Semantics

Our logic is an LTL-like logic with an epistemic operator. Let A be a name-variable-distinct process that $\mathrm{fv}(A) \setminus \mathrm{dom}(A) = \widetilde{x}$, ρ be an assignment which maps $\widetilde{x}$ to ground terms, $\mathbf{tr} \in \mathrm{tr}(A\rho)$, $0 \leq i \leq |\mathbf{tr}|$, and M_1 and M_2 be terms. Please remember that $\mathrm{fr}(A)$ is a frame of A.

We suppose that δ and φ contain no variables other than $\widetilde{x} \cup \mathrm{dom}(\mathbf{tr}[i])$. We omit semantics of logical operators. They are defined as expected.

$$\begin{aligned}
A, \rho, \mathbf{tr}, i &\models M_1 = M_2 \text{ iff } (M_1\rho = M_2\rho)\mathrm{fr}(\mathbf{tr}[i]) \\
A, \rho, \mathbf{tr}, i &\models M \in \mathrm{dom} \text{ iff } M \text{ is a variable } x, \text{ and } x \in \mathrm{dom}(\mathbf{tr}[i]) \\
A, \rho, \mathbf{tr}, i &\models \langle \mu \rangle_{-} \varphi \text{ iff } \mathbf{tr}[i-1] \xRightarrow{\mu} \mathbf{tr}[i] \text{ in } \mathbf{tr} \text{ and } A, \rho, \mathbf{tr}, i-1 \models \varphi \\
A, \rho, \mathbf{tr}, i &\models F\varphi \text{ iff } \exists j \geq i \text{ s.t. } A, \rho, \mathbf{tr}, j \models \varphi \\
A, \rho, \mathbf{tr}, i &\models K\varphi \text{ iff } \forall \rho' \forall \mathbf{tr}' \in \mathrm{tr}(A\rho'); \mathbf{tr}[0,i] \sim_t \mathbf{tr}'[0,i] \Rightarrow A, \rho', \mathbf{tr}', i \models \varphi
\end{aligned}$$

We suppose that an attacker does not know what terms are assigned to free variables before the process runs. Hence, the definition of K contains a quantifier over assignments $\forall \rho'$. Recall that an attacker can observe only labeled transitions, so accessibility is defined based on static equivalence on traces.

We also define satisfiability of formulas containing free variables. We put $\widetilde{y} = \mathrm{dom}(\mathbf{tr}[i])$. We suppose that φ contains no variables other than $\widetilde{x}, \widetilde{y}$, and $\widetilde{z}$.

$$A, \rho, \mathbf{tr}, i \models \varphi(\widetilde{x}, \widetilde{y}, \widetilde{z}) \text{ iff } \forall \widetilde{M}; A, \rho, \mathbf{tr}, i \models \varphi(\widetilde{x}, \widetilde{y}, \widetilde{M}),$$

where $\widetilde{M}$ is a sequence of ground terms.

From now, we suppose that all processes are name-variable-distinct. We often omit restriction of a domain of definition. $D(\rho)$ is a domain of definition of ρ.

When a formula φ is satisfied over all possible runs of a process A, we say that A satisfies φ.

Definition 8. $A \models \varphi \overset{\mathrm{def}}{\Leftrightarrow} \forall \rho\ \forall \mathbf{tr} \in \mathrm{tr}(A\rho); A, \rho, \mathbf{tr}, 0 \models \varphi$, *where* $D(\rho) = \mathrm{fv}(A) \setminus \mathrm{dom}(A)$.

Definition 9. $A \sqsubseteq_L B \overset{\mathrm{def}}{\Leftrightarrow} \forall \rho\ \forall \mathbf{tr} \in \mathrm{tr}(A\rho)\ \exists \mathbf{tr}' \in \mathrm{tr}(B\rho)$
s.t. $\forall i\ \forall \varphi; [A, \rho, \mathbf{tr}, i \models \varphi \Leftrightarrow B, \rho, \mathbf{tr}', i \models \varphi]$,
where $D(\rho) = (\mathrm{fv}(A) \setminus \mathrm{dom}(A)) \cup (\mathrm{fv}(B) \setminus \mathrm{dom}(B))$.
$A \equiv_L B \overset{\mathrm{def}}{\Leftrightarrow} A \sqsubseteq_L B$ *and* $B \sqsubseteq_L A$.

4.3 Correspondence with Trace Equivalence

We prove that trace equivalent processes satisfy the same formulas.

Theorem 2. *1.* $A \approx_t B \Rightarrow A \sqsubseteq_L B$*; 2.* $A \sqsubseteq_L B \Rightarrow A \subseteq_t B$

Proof. Let $\widetilde{x} = (\mathrm{fv}(A) \setminus \mathrm{dom}(A)) \cup (\mathrm{fv}(B) \setminus \mathrm{dom}(B))$.

1) We prove

$$\forall\rho\ \forall \mathbf{tr} \in \mathrm{tr}(A\rho), \mathbf{tr}' \in \mathrm{tr}(B\rho);$$
$$\mathbf{tr} \sim_t \mathbf{tr}' \Rightarrow \forall i\ \forall\varphi; [A, \rho, \mathbf{tr}, i \models \varphi \Leftrightarrow B, \rho, \mathbf{tr}', i \models \varphi]$$

where $D(\rho) = \widetilde{x}$, by induction on the syntax of formulas.

2) We arbitrarily take an assignment ρ and $\mathbf{tr} \in \mathrm{tr}(A\rho)$.
By $A \sqsubseteq_L B$, $\exists \mathbf{tr}' \in \mathrm{tr}(B\rho)$ s.t.$\forall i\ \forall\varphi; [A, \rho, \mathbf{tr}, i \models \varphi \Leftrightarrow B, \rho, \mathbf{tr}', i \models \varphi]$.
Then, we can prove $\mathbf{tr} \sim_t \mathbf{tr}'$.
By arbitrariness of $\mathbf{tr}$, it immediately follows that $A \sqsubseteq_L B \Rightarrow A \subseteq_t B$. □

Theorem 3. *$A \approx_t B \Leftrightarrow A \equiv_L B$.*

This theorem suggests that trace equivalence is suitable to define security properties. We give Proposition 2 as an example in the next subsection.

4.4 Applications

In this subsection, we often use abbreviations. Notably, P expresses $\neg K \neg$, and G expresses $\neg F \neg$. $P\varphi$ means that an attacker does not know φ does not hold. In other words, the attacker thinks that the possibility that φ holds remains.

We define minimal secrecy. We regard it as generalized minimal anonymity [25].

Definition 10. *A variable x is minimally secret with respect to δ in A iff $A \models G(\delta(x) \to P(\neg\delta(x)))$.*

This definition means that attackers cannot know that $\delta(x)$ holds.

This property is very weak. For instance, although x is minimally secret with respect to a nontrivial formula δ, x is not always minimally secret with respect to $\neg\delta$. Hereafter, we often omit objects.

Example 3. We put $\delta(z) : z \neq a\ \wedge z \neq b$.

We consider a process if $x = a$ then $\overline{c}$ else $\overline{d}$. Then x is minimally secret with respect to δ, but not secret with respect to $\neg\delta$.

Moreover, $\vee$ does not preserve minimally secret. However, $\wedge$ preserves it.

Although x is minimally secret in A, x is not always secret in $A|A$. Besides, restriction does not always preserve minimal secrecy.

Example 4. We put $\delta(z) : z = a$. We put

$$P = \text{if } x = a \text{ then } (\overline{a}\langle s\rangle + \overline{b}\langle s\rangle) \text{ else } \overline{a}\langle s\rangle, Q = \text{if } x = b \text{ then } \overline{b}\langle s\rangle \text{ else } \overline{c}\langle s\rangle.$$

Then x is minimally secret with respect to δ in $P + Q$, but not secret in $(P + Q)|(P + Q)$.

Example 5. We put $\delta(z) : z = a$. Then, x is minimally secret with respect to δ in $\overline{x} + \overline{a}$, but not secret in $\nu a.(\overline{x} + \overline{a})$. Here, we omitted objects.

We define total secrecy. We can also regard it as generalized total anonymity [25]. Total secrecy states attackers can obtain no information about x.

Definition 11. *x is totally secret in $A(x, \widetilde{y})$ iff*

$$\forall\delta(z, \widetilde{z}, \widetilde{w}); A(x, \widetilde{y}) \models G(\delta(x, \widetilde{y}, \widetilde{w}) \to P(\neg\delta(x, \widetilde{y}, \widetilde{w})))$$

where δ contains no variables other than ones in $\{z\} \cup \widetilde{z} \cup \widetilde{w}$ and satisfies that $\forall\widetilde{N}\forall\psi\exists M$: ground s.t. $\psi \models \neg\delta(M, \widetilde{N}, \widetilde{w})$. *Besides, $|\widetilde{y}| = |\widetilde{z}|$ and $\widetilde{w} \cap (\{x\} \cup \widetilde{y}) = \emptyset$.*

Proposition 2. *x is totally secret in $A(x, \widetilde{y})$ iff $A(x, \widetilde{y}) \approx_t A(x', \widetilde{y})$.*

Proof. $\Rightarrow$) We suppose for the sake of contradiction that $A(x, \widetilde{y}) \not\approx_t A(x', \widetilde{y})$.

There exist M_1, M_2 and $\widetilde{N}$ that are ground such that $A(M_1, \widetilde{N}) \not\approx_t A(M_2, \widetilde{N})$.

We suppose that $A(M_1, \widetilde{N}) \not\sqsubseteq_t A(M_2, \widetilde{N})$. Then, there exists $\mathbf{tr} \in \mathrm{tr}(A(M_1, \widetilde{N}))$ such that any trace of $A(M_2, \widetilde{N})$ is not statically equivalent to $\mathbf{tr}$.

We put $\delta(z, \widetilde{z}) : z \neq M_2 \vee \widetilde{z} \neq \widetilde{N}$. Then

$$A(x, \widetilde{y}), (x \mapsto M_1, \widetilde{y} \mapsto \widetilde{N}), \mathbf{tr}, |\mathbf{tr}| \models K\delta(x, \widetilde{y}).$$

This contradicts total secrecy.

$\Leftarrow$) We arbitrarily take $\delta, \rho, \mathbf{tr}$ and i, where δ meets the demand of Definition 11 and $D(\rho) = \{x\} \cup \widetilde{y}$.

We suppose that $A(x, \widetilde{y}), \rho, \mathbf{tr}, i \models \delta(x, \widetilde{y}, \widetilde{w})$.

We take M such that $\mathrm{fr}(\mathbf{tr}[i]) \models \neg\delta(M, \rho(\widetilde{y}), \widetilde{w})$. Let ρ' be

$$\rho'(y) = \begin{cases} M & (y = x) \\ \rho(y) & (otherwise). \end{cases}$$

By assumption, $A(\rho(x), \rho(\widetilde{y})) \approx_t A(M, \rho(\widetilde{y}))$.

Hence, there exists $\mathbf{tr}' \in \mathrm{tr}(A(M, \rho(\widetilde{y})))$ such that $\mathbf{tr} \sim_t \mathbf{tr}'$.

Then, $A(x, \widetilde{y}), \rho', \mathbf{tr}', i \models \neg\delta(M, \rho(\widetilde{y}), \widetilde{w})$.

Therefore, $A(x, \widetilde{y}), \rho, \mathbf{tr}, i \models P(\neg\delta(x, \rho(\widetilde{y}), \widetilde{w}))$.

Then, $A(x, \widetilde{y}) \models G(\delta(x, \widetilde{y}, \widetilde{w}) \to P(\neg\delta(x, \rho(\widetilde{y}), \widetilde{w})))$. □

Theorem 4. *If x is totally secret in $A(x, \widetilde{y})$, then x is also totally secret in $E[A(x, \widetilde{y})]$ for every context $E[\_]$ which does not contain x.*

Our framework can handle role interchangeability [25]. When x_i satisfies a property δ_k and x_l satisfies a property δ_j, an attacker thinks that it is possible that x_l satisfies a property δ_k and x_i satisfies a property δ_j.

Definition 12. *We put* $\mathrm{fv}(A) \setminus \mathrm{dom}(A) = \{x_1, ..., x_p\}$, *$J = \{1, ..., q\}$, and $I = \{1, ..., p\}$. (x_i, δ_k) is role interchangeable regarding $\{\delta_j(z_j, \widetilde{y_j})\}_{j \in J}$ in A iff*

$$A(x_1, ..., x_p) \models G(\delta_k(x_i, \widetilde{y_k}) \to \bigwedge_{l \in I} \bigwedge_{j \in J} (\delta_j(x_l, \widetilde{y_j}) \to P(\delta_k(x_l, \widetilde{y_k}) \wedge \delta_j(x_i, \widetilde{y_j}))))$$

where $\widetilde{y_j} \cap \{x_1, ..., x_p\} = \emptyset$ for all $j \in J$.

Proposition 3
$\forall \widetilde{M}\ \forall i\ \forall \mathbf{tr} \in \mathrm{tr}(A(M_1, ..., M_p))$
$\exists \widetilde{N}\ \exists \mathbf{tr}' \in \mathrm{tr}(A(M_i, N_2, ..., N_{i-1}, M_1, N_{i+1}, ..., N_p))$ s.t. $\mathbf{tr} \sim_t \mathbf{tr}'$
$\Leftrightarrow$ *(x_1, δ_k) is role interchangeable with respect to $\{\delta_j\}$ in A for all $\{\delta_j\}$ and k.*

Corollary 1. *$\forall l \in I \setminus \{i\}; A(x_1, ..., x_i, ..., x_l, ..., x_p) \approx_t A(x_1, ..., x_l, ..., x_i, ..., x_p)$ $\Rightarrow$ (x_i, δ_k) is role interchangeable with respect to $\{\delta_j\}$ in A for all $\{\delta_j\}$ and k.*

The converse holds only when $p = 2$. We give a counterexample for $p = 3$.

Example 6. We put $A(x, y, z) =$ if $x = y$ then $\overline{x} + \overline{z}$ else if $x = z$ then $\overline{x} + \overline{y}$ else $\overline{y} + \overline{z}$.

Then, (x, δ_k) is role interchangeable regarding $\{\delta_j\}$ in A for all $\{\delta_j\}_{j \in J}$ and k, but $A(a, b, a) \not\approx_t A(b, a, a)$. Thus, $A(x, y, z) \not\approx_t A(y, x, z)$.

We can also consider role permutativity. Mano [24] showed that it is strictly stronger than role interchangeability. Role permutativity states that even if p values are swapped, an attacker cannot notice it.

Definition 13. *We put $\mathrm{fv}(A) \setminus \mathrm{dom}(A) = \{x_1, ..., x_p\}$, $J = \{1, ..., q\}$, and $I = \{1, ..., p\}$. $\{\delta_j\}_{j \in J}$ is role permutable in A iff*

$$\forall n \leq p\ \forall \psi \in \mathfrak{S}_p; A(x_1, ..., x_p) \models G(\bigwedge_{k \leq n} \delta_{i_k}(x_{i_k}, \widetilde{y_k}) \to P(\bigwedge_{k \leq n} \delta_{i_k}(x_{i_{\psi(k)}}, \widetilde{y_k})))$$

where $\widetilde{y_j} \cap \{x_1, ..., x_p\} = \emptyset$ for all j and each i_k differs.

Proposition 4. *$\forall \psi \in \mathfrak{S}_p; A(x_1, ..., x_p) \approx_t A(x_{\psi(1)}, ..., x_{\psi(p)})$*
$\Leftrightarrow$ $\{\delta_j\}_{j \in J}$ is role permutable in A for all $\{\delta_j\}_{j \in J}$.

We define openness. We regard it as generalized identity [32]. Parallel composition does not preserve openness.

Definition 14. *x is open in A under $\Delta(x)$ iff*

$$\forall \rho\ \forall \mathbf{tr} \in \mathrm{tr}_{\max}(A\rho); A, \rho, \mathbf{tr}, |\mathbf{tr}| \models \Delta(x) \to K(\Delta(x) \to (x = x\rho)).$$

Example 7. We put $\Delta(z) : z = r \vee z = s$. We put

$$P = \text{if } x = r \text{ then } \overline{a}\langle n \rangle \text{ else } \overline{b}\langle n \rangle,\ Q = \text{if } x = r \text{ then } \overline{b}\langle n \rangle \text{ else } \overline{a}\langle n \rangle.$$

Then x is open in P and Q under $\Delta(x)$, but x is not open in $P|Q$ under $\Delta(x)$.

Problem 1
Input: An extended process A, an assignment ρ, a trace $\mathbf{tr} \in \mathrm{tr}(A)$, an integer $0 \leq i \leq |\mathbf{tr}|$, and a formula φ.
Question: Does $A, \rho, \mathbf{tr}, i \models \varphi$ hold?

Proposition 5. *Even if the word problem in Σ is decidable, Problem 1 can be undecidable.*

Abadi and Cortier [2] proved that static equivalence can be undecidable even if the word problem in Σ is decidable. Proposition 5 follows from it.

We change semantics. We repeat the definition of satisfaction.

$$A \models \varphi \text{ iff } \forall \rho\ \forall \mathbf{tr} \in \mathrm{tr}(A\rho); A, \rho, \mathbf{tr}, 0 \models \varphi$$

Now, we restrict ρ to be an assignment which maps free variables to only names and restrict inputted messages in **tr** to be only variables. That is, we assume that an attacker cannot tamper with a message. In other words, the attacker can only transfer messages without any change. Notably, the attacker cannot make a tuple of messages.

Problem 2
Input: An extended process A and a formula φ.
Question: Does $A \models \varphi$ hold?

A convergent subterm theory is an equational theory defined by finite equations whose each right-hand side is a proper subterm of the left-hand side.

Proposition 6. *If the equational theory on Σ is a convergent subterm theory and A contains no replications, Problem 2 is decidable.*

4.5 Comparison with the Work of Chadha et al.

Chadha et al. developed the definition of privacy in e-voting as follows. They considered protocol instances in which two voters Alice and Bob participate, and voting options are **0** and **1**.

Definition 15 ([7, Definition 9]). *The voting process $\mathcal{V}$ respects* privacy *if* $\mathcal{V} \models \mathbf{Aprivacy} \wedge \mathbf{Bprivacy}$ *where*

- $\mathbf{Aprivacy} \overset{\text{def}}{=} \wedge_{\mathrm{v} \in \{\mathbf{0},\mathbf{1}\}} \Box(\mathbf{K}(\mathbf{Avote}(\mathrm{v})) \to \mathbf{Bvote}(\mathrm{v}))$, *and*
- $\mathbf{Bprivacy} \overset{\text{def}}{=} \wedge_{\mathrm{v} \in \{\mathbf{0},\mathbf{1}\}} \Box(\mathbf{K}(\mathbf{Bvote}(\mathrm{v})) \to \mathbf{Avote}(\mathrm{v}))$.

$\mathbf{Avote}(\mathrm{v})$ means that Alice voted v, and $\mathbf{Bvote}(\mathrm{v})$ is similar.

Minimal secrecy of a vote never holds because an attacker trivially knows votes when all votes agree. We consider protocol instances in which m voters participate and n voting options exist. Let v_i be a vote of i. We consider the property below:

$$\vee_{j,k} v_j \neq v_k \to\ \wedge_i \wedge_v G(K(v_i = v) \to v_1 = v \wedge ... \wedge v_{i-1} = v \wedge v_{i+1} = v \wedge ... \wedge v_m = v)$$

The consequence in $G(...)$ implies that $v_i \neq v$ due to the antecedent condition, so we can rewrite the property.

$$\vee_{j,k} v_j \neq v_k \to \wedge_i \wedge_v G(K(v_i = v) \to v_i \neq v)$$

Moreover, we take the contraposition in G.

$$\vee_{j,k} v_j \neq v_k \to \wedge_i \wedge_v G(v_i = v \to P(v_i \neq v))$$

This consequence is exactly minimal secrecy. Besides, minimal secrecy of voting implies privacy, so privacy and minimal secrecy of voting agree under the disagreement condition $\vee_{j,k} v_j \neq v_k$.

It was shown that $V(\mathbf{0},\mathbf{1}) \approx_t V(\mathbf{1},\mathbf{0})$ implies that $\mathcal{V}$ respects privacy, and the partial converse was given in [7]. We give several properties of minimal secrecy.

Proposition 7. *We assume that a voting process $\mathcal{V}$ is equivalent for aborts, and minimal secrecy of each vote in $V(v_1,...,v_m)$ holds under the disagreement condition $\vee_{j,k} v_j \neq v_k$.*

1. $m = 2 \Rightarrow V(v_1,v_2) \approx_t V(v_2,v_1)$.
2. $m = 3$ *and* $n = 2 \Rightarrow V(v_1,v_2,v_3) \approx_t V(v_2,v_1,v_3)$.
3. *Otherwise,* $V(v_1,...,v_i,...,v_j,...,v_m) \approx_t V(v_1,...,v_j,...,v_i,...,v_m)$ *does not always hold.*

5 Related Work

Logics about behavior of labeled transition systems originate from Hennessy-Milner logic [20] that is a modal logic characterizing observational congruence. That is, observational equivalent systems satisfy the same modal formulas when these systems are image-finite.

Process algebra is a special LTS. The spi calculus [3] is an extension of the pi-calculus. It enables us to handle symmetric key encryption based on the Dolev-Yao model [14]. In the spi calculus, two ciphertexts obtained by encrypting two different plaintexts are indistinguishable unless an observer gets a secret key. Abadi and Gordon formalized security properties to use testing equivalence.

We focused on the applied pi calculus [1] because it is more powerful than the spi calculus. That is, we intend to handle more various security notions. In the calculus, a process can send not only names but also terms via alias variables. Due to this feature, we can handle not only secrecy but also stricter properties. The authors proved that observational equivalence and labeled bisimilarity correspond.

Chadha et al. [7] already developed an epistemic logic for the applied pi calculus. They defined formulas **Has** and $\widehat{\mathbf{evt}}$. **Has** directly represents attackers' knowledge, and $\widehat{\mathbf{evt}}$ means that a particular event had occurred. Temporal modalities were also used, but they do neither mention the just previous nor next action. The epistemic operator **K** was defined based on static equivalence on traces. Authors suggested that trace equivalence is more suitable than labeled bisimilarity when we consider privacy. However, a correspondent relation between logic and behavior of processes was not provided. As a matter of fact, α-equivalent processes do not always satisfy the same formulas in their framework because secret values are expressed as bound names or through events. In our framework, trace equivalent processes satisfy the same formulas.

Horne [21] introduced quasi-open bisimilarity, and he proved that it coincides open bisimilarity. Moreover, intuitionistic modal logic $\mathcal{FM}$ characterizes quasi-open bisimilarity. The law of excluded middle does not hold in the logic because processes containing a free variable are also considered.

Knight et al. [22] defined an epistemic logic for an LTS. This framework is based on Hennessy-Milner logic, and it handles multiple agents' knowledge. They also proved weak completeness. However, compositionality was not discussed.

Process algebra is one of nominal transition systems. Parrow et al. [29] developed modal logic characterizing bisimilarity for a nominal transition system.

Toninho and Caires [31] proposed a dynamic spatial epistemic logic, which reasons what information a process can obtain. The epistemic operator means not only an attacker's knowledge but also a participant's knowledge, so, for example, the logic can reason a correspondence assertion.

Tsukada et al. [32] studied sequential and parallel compositionality of security notions to use an epistemic logic for a multiagent system. They proved that neither anonymity nor privacy is generally preserved by composition. They also gave a sufficient condition for preservation. However, this word "parallel" merely means that the same agent acts two actions in the paper. That is, concurrency was not considered.

Fiore and Abadi [16] developed symbolic models of processes. They gave a procedure to decide whether an environment can derive a message M. Their technique can be used for verification. However, equivalences on processes were not studied in the paper.

Clarkson and Schneider [12] generalized trace properties to hyperproperties, and Clarkson et al. [11] developed hyperLTL and hyperCTL* for hyperproperties. Hyperproperties can express security properties which cannot be expressed by trace properties. The authors regarded systems as sets of traces, so hyperproperties are properties about systems. Our security properties are also proper hyperproperties. The advantage of our work over these works is to relate trace equivalence to attackers' knowledge. In [11,12], the relation between the equivalence and knowledge is not clear.

Goubault-Larrecq et al. [19] proposed the probabilistic applied pi calculus. In this case, Theorem 1 no longer holds. It is known that trace distribution preorder [30] is not a congruence. On the other hand, it is shown in [5] that probabilistic trace equivalence for nondeterministic and probabilistic LTS is a congruence with respect to parallel composition. Probabilistic trace equivalence is coarser than trace distribution equivalence.

Canetti et al. [6] defined implementation for task-PIOAs. According to their definition, $\mathcal{T}_1$ implements $\mathcal{T}_2$ iff the set of behaviors of $\mathcal{T}_1$ composed with $\mathcal{E}$ is included in the set of behaviors of $\mathcal{T}_2$ composed with $\mathcal{E}$ for every environment $\mathcal{E}$. Here, behavior is the set of trace distributions. The implementation relation is preserved by parallel composition.

Giro and D'Argenio [17] pointed out that ordinary schedulers may give rise to unnatural behavior. A scheduler may leak information if it can look at the whole of the system. To solve this problem, they provided several reasonable subclasses of schedulers. The problem of the scheduler in the formalization of security properties was also pointed out in [4,8], which proposed other approaches.

Eisentraut et al. [15] also studied subclasses of schedulers for probabilistic automata. They defined late distribution bisimulation and proved that late distribution bisimulation with respect to distributed schedulers is compositional. We may need to specify subclasses of schedulers to state a probabilistic variant of Theorem 1.

Knight et al. [23] developed spatial and epistemic process calculus. Their study is for concurrent constraint programming, so their processes can add constraints. They proved that observational equivalence is a congruence. Their processes do not have labeled actions, so observational equivalence means that equivalent processes provide the same results. On the other hand, in the applied pi calculus, trace equivalent processes provide equivalent traces and indistinguishable information.

In this paper, we characterized trace equivalence in terms of our epistemic logic. That is, we showed that a non-adaptive active intruder cannot distinguish trace equivalent processes. We also focused on how composition of systems affects security properties. We proved that any composition preserves total secrecy and role permutativity. This is because trace equivalence is a congruence.

6 Conclusion

6.1 Summary

In this paper, we provided an epistemic logic for the applied pi calculus. This logic is an LTL-like logic, so we can describe security notions. We formulated secrecy, role-interchangeability, and openness. These are generalized security properties regarding multiagent systems.

Moreover, we associated trace equivalence with total secrecy. Application of context does not preserve minimal secrecy, but total secrecy is preserved because trace equivalence is a congruence. We also give a necessary and sufficient condition for role-interchangeability.

We conclude that trace equivalence is suitable to express non-probabilistic indistinguishability in the view of security in the presence of a non-adaptive active adversary.

6.2 Future Work

First, our epistemic logic states an adversary's knowledge. We intend to construct a logic for a process's knowledge. It will bridge a gap between multiagent systems and process calculi.

Second, formalizations of other security properties such as non-malleability are also the next topics.

Finally, what logic is suitable for security in the presence of an adaptive attacker is still open.

Acknowledgments. The author thanks Prof. M. Hasegawa and Prof. N. Yoshida for discussions. He also thanks the anonymous reviewers for helpful comments and suggestions.

References

1. Abadi, M., Blanchet, B., Fournet, C.: The applied pi calculus: mobile values, new names, and secure communication. J. ACM **65**(1), 1–41 (2017). https://doi.org/10.1145/3127586
2. Abadi, M., Cortier, V.: Deciding knowledge in security protocols under equational theories. Theor. Comput. Sci. **367**(1–2), 2–32 (2006). https://doi.org/10.1016/j.tcs.2006.08.032
3. Abadi, M., Gordon, A.D.: A calculus for cryptographic protocols: the spi calculus. Inf. Comput. **148**(1), 1–70 (1999). https://doi.org/10.1006/inco.1998.2740
4. Alvim, M.S., Andrés, M.E., Palamidessi, C., van Rossum, P.: Safe equivalences for security properties. In: Calude, C.S., Sassone, V. (eds.) TCS 2010. IAICT, vol. 323, pp. 55–70. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15240-5_5
5. Bernardo, M., De Nicola, R., Loreti, M.: Revisiting trace and testing equivalences for nondeterministic and probabilistic processes. Log. Methods Comput. Sci. **10**(1) (2014). https://doi.org/10.2168/LMCS-10(1:16)2014
6. Canetti, R., Cheung, L., Kaynar, D., Liskov, M., Lynch, N., Pereira, O., Segala, R.: Task-structured probabilistic i/o automata. J. Comput. Syst. Sci. **94**, 63–97 (2018). https://doi.org/10.1016/j.jcss.2017.09.007
7. Chadha, R., Delaune, S., Kremer, S.: Epistemic logic for the applied pi calculus. In: Lee, D., Lopes, A., Poetzsch-Heffter, A. (eds.) FMOODS/FORTE -2009. LNCS, vol. 5522, pp. 182–197. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02138-1_12
8. Chatzikokolakis, K., Palamidessi, C.: Making random choices invisible to the scheduler. In: Caires, L., Vasconcelos, V.T. (eds.) CONCUR 2007. LNCS, vol. 4703, pp. 42–58. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74407-8_4
9. Cheval, V., Kremer, S., Rakotonirina, I.: DEEPSEC: deciding equivalence properties in security protocols theory and practice. In: SP 2018, pp. 529–546 (2018). https://doi.org/10.1109/SP.2018.00033
10. Cheval, V., Cortier, V., Delaune, S.: Deciding equivalence-based properties using constraint solving. Theor. Comput. Sci. **492**, 1–39 (2013). https://doi.org/10.1016/j.tcs.2013.04.016
11. Clarkson, M.R., Finkbeiner, B., Koleini, M., Micinski, K.K., Rabe, M.N., Sánchez, C.: Temporal logics for hyperproperties. In: Abadi, M., Kremer, S. (eds.) POST 2014. LNCS, vol. 8414, pp. 265–284. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54792-8_15
12. Clarkson, M.R., Schneider, F.B.: Hyperproperties. J. Comput. Secur. **18**(6), 1157–1210 (2010). https://doi.org/10.3233/JCS-2009-0393
13. Delaune, S., Kremer, S., Ryan, M.: Verifying privacy-type properties of electronic voting protocols. J. Comput. Secur. **17**(4), 435–487 (2009). https://doi.org/10.3233/JCS-2009-0340
14. Dolev, D., Yao, A.C.: On the security of public key protocols. IEEE Trans. Inf. Theory **29**(2), 198–208 (1983). https://doi.org/10.1109/TIT.1983.1056650
15. Eisentraut, C., Godskesen, J.C., Hermanns, H., Song, L., Zhang, L.: Probabilistic bisimulation for realistic schedulers. In: Bjørner, N., de Boer, F. (eds.) FM 2015. LNCS, vol. 9109, pp. 248–264. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19249-9_16
16. Fiore, M., Abadi, M.: Computing symbolic models for verifying cryptographic protocols. In: CSFW-14, pp. 160–173 (2001). https://doi.org/10.1109/CSFW.2001.930144

17. Giro, S., D'Argenio, P.: On the expressive power of schedulers in distributed probabilistic systems. Electron. Notes Theor. Comput. Sci. **253**(3), 45–71 (2009). https://doi.org/10.1016/j.entcs.2009.10.005
18. Glabbeek, R.J.: The linear time - branching time spectrum. In: Baeten, J.C.M., Klop, J.W. (eds.) CONCUR 1990. LNCS, vol. 458, pp. 278–297. Springer, Heidelberg (1990). https://doi.org/10.1007/BFb0039066
19. Goubault-Larrecq, J., Palamidessi, C., Troina, A.: A probabilistic applied pi–calculus. In: Shao, Z. (ed.) APLAS 2007. LNCS, vol. 4807, pp. 175–190. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76637-7_12
20. Hennessy, M., Milner, R.: On observing nondeterminism and concurrency. In: de Bakker, J., van Leeuwen, J. (eds.) ICALP 1980. LNCS, vol. 85, pp. 299–309. Springer, Heidelberg (1980). https://doi.org/10.1007/3-540-10003-2_79
21. Horne, R.: A bisimilarity congruence for the applied pi-calculus sufficiently coarse to verify privacy properties. arXiv:1811.02536 (2018)
22. Knight, S., Mardare, R., Panangaden, P.: Combining epistemic logic and hennessy-milner logic. In: Constable, R.L., Silva, A. (eds.) Logic and Program Semantics. LNCS, vol. 7230, pp. 219–243. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29485-3_14
23. Knight, S., Palamidessi, C., Panangaden, P., Valencia, F.D.: Spatial and epistemic modalities in constraint-based process calculi. In: Koutny, M., Ulidowski, I. (eds.) CONCUR 2012. LNCS, vol. 7454, pp. 317–332. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32940-1_23
24. Mano, K.: Formal specification and verification of anonymity and privacy. Ph.D. thesis, Nagoya University (2013)
25. Mano, K., Kawabe, Y., Sakurada, H., Tsukada, Y.: Role interchange for anonymity and privacy of voting. J. Logic Comput. **20**(6), 1251–1288 (2010). https://doi.org/10.1093/logcom/exq013
26. Milner, R., Parrow, J., Walker, D.: A calculus of mobile processes, I. Inf. Comput. **100**(1), 1–40 (1992). https://doi.org/10.1016/0890-5401(92)90008-4
27. Milner, R., Parrow, J., Walker, D.: A calculus of mobile processes, II. Inf. Comput. **100**(1), 41–77 (1992). https://doi.org/10.1016/0890-5401(92)90009-5
28. Minami, K.: Trace equivalence and epistemic logic to express security properties. arXiv:1903.03719 (2019)
29. Parrow, J., Borgström, J., Eriksson, L.H., Gutkovas, R., Weber, T.: Modal logics for nominal transition systems. In: CONCUR 2015. LIPIcs, vol. 42, pp. 198–211. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2015). https://doi.org/10.4230/LIPIcs.CONCUR.2015.198
30. Segala, R.: A compositional trace-based semantics for probabilistic automata. In: Lee, I., Smolka, S.A. (eds.) CONCUR 1995. LNCS, vol. 962, pp. 234–248. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-60218-6_17
31. Toninho, B., Caires, L.: A spatial-epistemic logic for reasoning about security protocols. In: SecCo 2010. EPTCS, vol. 51, pp. 1–15. Open Publishing Association (2011). https://doi.org/10.4204/EPTCS.51.1
32. Tsukada, Y., Sakurada, H., Mano, K., Manabe, Y.: On compositional reasoning about anonymity and privacy in epistemic logic. Ann. Math. Artif. Intell. **78**(2), 101–129 (2016). https://doi.org/10.1007/s10472-016-9516-8