Математическая модель и методы верификации криптографических протоколов

А. М. Миронов[1]

В настоящей работе излагается новая математическая модель криптографических протоколов, и приводятся примеры применения этой модели для решения задач верификации криптографических протоколов. Криптографические протоколы – это распределенные алгоритмы, предназначенные для обеспечения передачи конфиденциальной информации в небезопасной среде. Они используются, например, в электронных платежах, электронных процедурах голосования, системах доступа к конфиденциальным данным, и т.д. Ошибки в криптографических протоколах могут привести к большому ущербу, поэтому необходимо использовать математические методы для обоснования различных свойств корректности и безопасности криптографических протоколов. В работе излагаются новые методы формальной верификации криптографических протоколов.

Ключевые слова: криптографические протоколы, последовательные процессы, распределенные процессы, верификация.

1. Введение

1.1. Понятие криптографического протокола

Криптографический протокол (КП) представляет собой распределенный алгоритм, описывающий порядок обмена сообщениями между несколькими агентами. Примеры таких агентов – компьютерные системы, банковские карточки, люди, и т.д.

Для обеспечения свойств безопасности КП (таких например как конфиденциальность передаваемых данных) в КП могут использоваться криптографические преобразования (шифрование, электронная подпись, хэш-функции, и т.п.). Мы предполагаем, что криптографические преобразования, используемые в КП, являются идеальными, т.е. удовлетворяют некоторым аксиомам, выражающим, например, невозможность

[1] *Миронов Андрей Михайлович* — доцент каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: amironov66@gmail.com.

Mironov Andrew Mikhailovich — associate professor, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

извлечения открытых текстов из шифртекстов без знания соответствующих криптографических ключей.

1.2. Уязвимости в криптографических протоколах

Многие уязвимости в КП связаны не с плохими криптографическими качествами используемых в них криптографических примитивов, а с логическими ошибками в КП. Наиболее ярким примером уязвимости в КП является уязвимость в КП аутентификации Нидхэма-Шредера [78NS], который был опубликован в 1978 г., и использовался в критических по безопасности информационных системах. Спустя более 16 лет после начала использования этого КП в нем обнаружилась логическая ошибка [95L], связанная с возможностью непредусмотренного нечестного поведения одного из участников этого КП и подрывающая безопасность этого КП. Особенность этой ошибки заключается в том, что данный КП является предельно простым распределенным алгоритмом, состоящим всего из трех действий, и при визуальном анализе этого КП отсутствие в нем ошибок не вызывало никаких сомнений. Ошибка была обнаружена лишь при помощи инструмента автоматизированной верификации КП.

Другой пример логической ошибки в КП (взят из статьи [14CK]): в КП входа в портал Google, позволяющем пользователю идентифицировать себя только один раз, а затем обращаться к различным приложениям (таким, например, как Gmail или календарь Google), обнаружена логическая ошибка, позволяющая нечестному поставщику услуг выдавать себя за любого из своих пользователей для другого поставщика услуг.

Существует много других примеров КП (см. например [81DS], [87NS], [95AN] [08CJSTW]), в которых обнаружились уязвимости следующего вида:

- участники этих КП могут получать искаженные сообщения (или вообще терять их) в результате перехвата, удаления или искажения противником передаваемых сообщений, что нарушает свойство целостности передаваемых сообщений,

- противник может узнать секретную информацию, содержащуюся в перехваченных сообщениях, в результате чего нарушается свойство конфиденциальности передаваемых сообщений.

Также есть примеры уязвимостей в КП, используемых для аутентификации перед провайдерами мобильной телефонной связи, для снятия денег в банкомате, для работы с электронными паспортами, проведения электронных выборов, и т.д.

Все эти примеры являются обоснованием того, что в критических по безопасности системах недостаточно неформального анализа требуемых свойств безопасности используемых в них КП, необходимо

- построение **математических моделей** анализируемых КП,
- описание свойств анализируемых КП в виде математических объектов, называемых **спецификациями** свойств этих КП, и
- построение формальных доказательств утверждений о том, что анализируемые КП удовлетворяют (или не удовлетворяют) своим спецификациям, процедура построения таких доказательств называется **верификацией** анализируемых КП.

В настоящей работе строится новая математическая модель КП, в терминах которой можно выражать такие свойства корректности КП, как например целостность и конфиденциальность передаваемых сообщений (т.е. обоснование следующих свойств анализируемого КП: сообщения, посланные одним участником этого КП другому участнику этого КП, доходят до получателя в неискаженном виде, и содержание этих сообщений не будет известно противнику), или аутентификация (т.е. доказательство подлинности) участников КП.

1.3. Основные методы моделирования и верификации криптографических протоколов

Обзоры наиболее широко используемых методов моделирования и верификации КП содержатся в книгах [11СК] и [12СМ]. Основные классы моделей КП и подходов к верификации КП имеют следующий вид.

1) **Логические модели**.

   Данный класс моделей был самым первым подходом к моделированию и верификации КП. На основе данного класса моделей проблема верификации КП сводится к проблеме построения в некотором логическом исчислении доказательства теоремы о том, что анализируемый КП обладает заданными свойствами. В работе [90BAN] была изложена первая математическая модель КП, называемая **логикой BAN** (название этой логики соответствует фамилиям ее создателей – Бэрроуза, Абади и Нидхэма). Данная модель имеет большие ограничения: в ней предполагается, что участники анализируемого КП являются честными, т.е. точно выполняют предписания КП. Такое ограничение не позволило обнаружить упомянутую выше уязвимость в КП Нидхэма-Шредера. Кроме того, данная модель не позволяет анализировать КП с неограниченным порождением сеансов. Аппарат логики BAN был развит в работах [90GNY],

[91AT], [93vO] [93SM], [94KMM], [96SvO], [02SW]. Важным классом логических исчислений для моделирования и анализа КП является композиционная логика протоколов (Protocol Composition Logic), которой посвящены работы [01DMP], [07DDMR], [08C], [11DMRS].

Одним из классов логических моделей КП связан с логическим программированием. В данных моделях шаги протокола представляются в виде правил переписывания термов. Для моделирования КП используются клаузы Хорна и системы уравнений с ограничениями (constraint systems). Данный подход излагается в работах [01B], [05AB], [14CK] и др.

Важным классом логических методов моделирования и анализа КП является индуктивный метод Паульсона: [97P], [98P], [99P], [00B].

2) Модели, основанные на **алгебре процессов**.

Источником данного класса моделей является основополагающая работа Р.Милнера [80M]. В данной работе строится модель взаимодействующих процессов, в которой процессы представляются термами. На этих термах вводится отношение наблюдаемой эквивалентности, которое позволяет эффективно выражать различные свойства процессов, связанные с безопасностью (в частности свойства секретности и анонимности). Первой работой, в которой излагается модель КП на базе подхода Р.Милнера, является статья М.Абади и А.Гордона [99AG]. Среди других работ, относящихся к этому направлению, можно отметить работы [00RS], [01AF], [05KR], [07ABF], [11RS], [16ABF], [16B], [17CW], [21CDS].

3) Модели, основанные на **CSP**.

CSP (**Communicating Sequential Processes**) – это математический аппарат, разработанный А.Хоаром [85H] и предназначенный для моделирования и анализа распределенных вычислительных процессов. На базе этого аппарата построен метод моделирования и верификации КП, наиболее полно изложенный в книге [00RSGLR]. Дедуктивная верификация КП на основе данного подхода использует понятие **ранг-функции**. Среди работ, относящихся к данному направлению, можно отметить работы [96SS], [96S], [97LR], [97DS], [98S], [21RCSSS].

4) Модели, основанные на **пространствах нитей** (**strand spaces**).

Пространства нитей позволяют представлять процессы, входящие в КП, в виде графических объектов (называемых нитями), в которых указаны зависимости между действиями, относящимися к различным процессам. Среди работ, относящихся к методам модели-

рования и верификации КП на основе понятия пространства нитей, можно отметить работы [98THG1], [98THG2], [99THG1], [99THG2], [00GT2], [02GT], [05CDLMS], [07DGT1], [07DGT2], [07DGT3], [12G], [13LP], [16YEMM].

1.4. Сравнение предлагаемой модели криптографических протоколов с другими моделями

Модель КП, излагаемая в настоящей работе, унаследовала наиболее существенные качества моделей каждого из перечисленных выше четырех классов. В этой модели КП представляются в виде распределенных процессов (РП), взаимодействующих путем асинхронной передачи сообщений через каналы. Каждый РП, соответствующий какому-либо КП, представляет собой совокупность последовательных процессов (ПП), моделирующих работу участников этого КП. Как правило,

- эти ПП представляют собой последовательности действий, которые графически можно изобразить в виде нитей, и
- выполнение всего КП можно представить в виде пространства нитей, точки на которых связаны ребрами, изображающими передачу и прием сообщений, см. например (68) и (95).

Свойства КП могут представляться в виде логических формул, для обоснования которых могут использоваться стандартные алгоритмы логического вывода. Кроме того, некоторые свойства КП (например анонимность) м.б. выражены в виде отношения наблюдаемой эквивалентности между соответствующими РП, аналогично тому, как это делается в моделях КП основанных на процессной алгебре.

Перечислим основные достоинства предложенной модели.

1) Доказательства свойств корректности КП на базе данной модели, существенно короче, чем доказательства этих свойств на базе других моделей КП. Для обоснования этого утверждения мы приводим примеры верификации двух КП: Yahalom [00RSGLR] и КП передачи сообщений, взятый из [99AG]. Верификация этих КП в вышеприведенных источниках занимает несколько десятков страниц, в то время как верификация КП Yahalom на базе предложенной модели (пункты 3.2.3, 3.2.4, 3.2.5) занимает менее 4 страниц, а верификация второго КП (пункт 3.3.2) – менее 3 страниц. Кроме того, анализ доказательств корректности данных КП показывает, что эти доказательства производятся по шаблонной методике и м.б. порождены автоматически.

2) Если анализируемые КП состоят из конечного числа компонентов без циклов, то верификация таких КП может быть проведена полностью автоматически, на основе введенного в настоящей работе понятия графа переходов, что показано на четырех примерах КП (пункты 3.1.4, 3.1.5, 3.1.7, 3.1.8), взятых из работы [99AG]. В этой работе верификация данных КП представляет собой нетривиальные математические рассуждения, в то время как в настоящей работе данные КП верифицируются путем вычисления формул, истинных в вершинах графов переходов.

3) Введенный в настоящей работе язык описания РП позволяет строить такие модели КП, которые имеют существенное сходство с исходными описаниями КП. Это обстоятельство является существенным в том случае, когда в анализируемом КП в результате верификации обнаружена ошибка, и необходимо так модифицировать анализируемый КП, чтобы устранить эту ошибку. Если модель КП схожа с описанием этого КП на исходном языке, то для устранения обнаруженной ошибки в КП сначала может быть выполнена коррекция модели этого КП, которая затем несложно преобразуется в коррекцию анализируемого КП на исходном языке.

   Отметим, что изложенный в настоящей работе язык описания РП имеет самостоятельную ценность, и может рассматриваться как новый язык описания распределенных алгоритмов.

2. Последовательные и распределенные процессы

В этом параграфе мы излагаем понятия последовательного и распределенного процессов. Последовательный процесс является моделью участника КП, а распределенный процесс является моделью всего КП. Предложенная модель является теоретической основой для методов верификации КП, излагаемых в параграфе 3.

2.1. Вспомогательные понятия

2.1.1Типы, константы, переменные, функциональные символы

Предполагаем, что заданы множества $Types$, Con, Var и Fun, элементы которых называются **типами**, **константами**, **переменными**, и **функциональными символами (ФС)**, соответственно.

Каждому элементу x множеств Con, Var и Fun сопоставлен некоторый тип $\tau(x) \in Types$, причем если $x \in Fun$, то $\tau(x)$ имеет вид

$$(\tau_1, \ldots, \tau_n) \to \tau, \quad \text{где } \tau_1, \ldots, \tau_n, \tau \in Types.$$

2.1.2. Термы

В этом пункте определяется множество Tm **термов**, которые предназначены для описания сообщений, пересылаемых во время выполнения КП. Множество Tm определяется индуктивно. Каждому терму e сопоставлен некоторый тип $\tau(e) \in Types$. Определение терма имеет следующий вид:

- $\forall\, x \in Con \cup Var \;\; x$ является термом типа $\tau(x)$,
- если $f \in Fun$, $e_1, \ldots, e_n$ – термы, и $\tau(f) = (\tau(e_1), \ldots, \tau(e_n)) \to \tau$, то запись $f(e_1, \ldots, e_n)$ является термом типа τ.

Будем использовать следующие обозначения:

- $\forall\, e \in Tm \;\; Var(e) = \{x \in Var \mid x \text{ входит в } e\}$,
- $\forall\, X \subseteq Var \;\; Tm(X) = \{e \in Tm \mid Var(e) \subseteq X\}$,
- $\forall\, E \subseteq Tm \;\; E_{\mathbf{X}} = E \cap Var$, и $\forall\, \tau \in Types \;\; E_\tau = \{e \in E \mid \tau(e) = \tau\}$.

Пусть $e, e' \in Tm$. Терм e называется **подтермом** терма e', если $e = e'$, или e' имеет вид $f(e_1, \ldots, e_n)$, где $f \in Fun$, и $\exists\, i \in \{1, \ldots, n\}$: e – подтерм терма e_i. Запись $e \subseteq e'$, где $e, e' \in Tm$, означает, что e – подтерм e'. Запись $e \subset e'$, где $e, e' \in Tm$, означает, что $e \subseteq e'$ и $e \neq e'$.

Индукцией по структуре терма $e \in Tm$ нетрудно доказать, что

$$\begin{array}{l}\text{если } e_1 \text{ и } e_2 \text{ – различные подтермы терма } e, \text{ то либо } e_1 \subset e_2,\\ \text{либо } e_2 \subset e_1, \text{ либо } e_1 \text{ и } e_2 \text{ не имеют общих компонентов.}\end{array} \qquad (1)$$

Запись $x \in e$, где $x \in Var, e \in Tm$ обозначает утверждение $x \subseteq e$.

Ниже для каждой рассматриваемой функции вида $f : E \to E'$, где $E, E' \subseteq Tm$, будем предполагать, что $\forall\, e \in E \;\; \tau(f(e)) = \tau(e)$.

2.1.3. Примеры типов

Будем считать, что $Types$ содержит следующие типы:

- тип **A**, термы этого типа называются **агентами**,
- тип **C**, термы этого типа называются **каналами**, они обозначают каналы связи, при помощи которых агенты взаимодействуют друг с другом путем передачи сообщений,

- тип $\mathbf{K}$, термы этого типа называются **ключами**, они обозначают криптографические ключи, которые агенты могут использовать для шифрования или дешифрования сообщений,
- тип $\mathbf{M}$, термы этого типа называются **сообщениями**, они обозначают сообщения, которые агенты могут пересылать друг другу во время своей работы,
- тип $\mathbf{N}$, термы этого типа называются **нонсами**, они обозначают переменные с уникальными значениями,
- тип $\mathbf{P}$, термы этого типа называются **процессами**.

Записи *Agents*, *Channels*, *Keys*, *Messages*, *Nonces* и *Processes* обозначают множества всех агентов, каналов, ключей, сообщений, нонсов и процессов, соответственно.

Будем использовать следующие соглашения и обозначения:

- множество *Channels* содержит переменную, обозначаемую символом $\circ$, и называемую **открытым каналом**,
- тип $\mathbf{M}$ включает все другие типы из $Types$, т.е. терм любого типа является также термом типа $\mathbf{M}$,
- $\forall\, n \geq 1, \forall\, \tau \in Types$ множество $Types$ содержит тип τ_n, значения которого – кортежи длины n, состоящие из значений типа τ.

2.1.4. Примеры функциональных символов

Будем предполагать, что Fun содержит следующие ФС.

- ФС $tuple_n$, где $n \geq 1$ и $\tau(tuple_n) = (\underbrace{\mathbf{M}, \ldots, \mathbf{M}}_{n}) \to \mathbf{M}_n$.

  Для каждого списка $(e_1, \ldots, e_n)$ термов терм $tuple_n(e_1, \ldots, e_n)$ будет обозначаться более короткой записью $(e_1, \ldots, e_n)$.

- ФС $pr_{n,i}$, где $n \geq 1$, $i \in \{1, \ldots, n\}$, и $\tau(pr_{n,i}) = \mathbf{M}_n \to \mathbf{M}$.

  $\forall\, e \in Tm_{\mathbf{M}_n}$ терм $pr_{n,i}(e)$ является i–й компонентой кортежа e, и будет обозначаться записью $(e)_i$.

- ФС $hash\_function$ (возможно с индексами) типа $\mathbf{M} \to \mathbf{M}$.

  Терм вида $hash\_function(e)$ обозначает значение **хэш-функции** сообщения e.

- ФС *encrypt* и *decrypt* типа $(\mathbf{K}, \mathbf{M}) \to \mathbf{M}$.

  Термы вида $encrypt(k, e)$ и $decrypt(k, e)$ обозначают сообщения, получаемые шифрованием (и дешифрованием, соответственно) сообщения e на ключе k.

- ФС $public\_key$ типа $\mathbf{A} \to \mathbf{K}$.

  Терм $public\_key(A)$ называется **открытым ключом** агента A.

  Термы вида $encrypt(k, e)$ и $encrypt(public\_key(A), e)$ будет обозначаться записями $k(e)$ и $A(e)$ соответственно, данные термы называются **шифрованными сообщениями (ШС)**.

- ФС $shared\_key$ типа $\mathbf{A}_n \to \mathbf{K}$, где $n \geq 2$ (т.е. одно и то же обозначение $shared\_key$ используется для семейства ФС).

  Терм вида $shared\_key(A_1, \ldots, A_n)$ называется **разделяемым ключом** агентов $A_1, \ldots, A_n$ и будет обозначаться записью $k_{A_1 \ldots A_n}$.

- ФС $shared\_channel$ типа $\mathbf{A}_n \to \mathbf{C}$, где $n \geq 2$ (т.е. одно и то же обозначение $shared\_channel$ используется для семейства ФС).

  Терм вида $shared\_channel(A_1, \ldots, A_n)$ называется **разделяемым каналом** агентов $A_1, \ldots, A_n$ и будет обозначаться записью $c_{A_1 \ldots A_n}$.

- ФС $digital\_signature$ типа $(\mathbf{M}, \mathbf{A}) \to \mathbf{M}$.

  Терм вида $digital\_signature(e, A)$ обозначает **цифровую подпись** сообщения e, сделанную агентом A.

  Тройка $(e, A, digital\_signature(e, A))$ будет обозначаться $(e)_A$.

Будем использовать следующие обозначения: $\forall\, e \in Tm$

$$\begin{aligned} VarEncKeys(e) &= \{k \in Var_{\mathbf{K}} \mid \exists\, e' \in Tm : k(e') \subseteq e\}, \\ OpenEncKeys(e) &= \{A \in Var_{\mathbf{A}} \mid \exists\, e' \in Tm : A(e') \subseteq e\}. \end{aligned}$$

2.1.5. Выражения

В этом пункте определяется множество $Expr$ **выражений**, которые предназначены для описания множеств термов. Например, в качестве такого множества может выступать совокупность термов, доступных в текущий момент какому-либо процессу, или совокупность сообщений, находящихся в текущий момент в каком-либо канале.

Выражением называется запись одного из следующих видов:

- E, где $E \subseteq Tm$,
- $[P]$ и $[c]$, где $P \in Processes$, $c \in Channels$,

- $k^{-1}(E)$, где $k \in Keys$, и $E \in Expr$,
- $E \cap E'$, $E \cup E'$, $\neg E$, где $E, E' \in Expr$.

$\forall\, E \in Expr \;\; Var(E) = \{x \in Var \mid x \text{ входит в } E\}$.

Выражения вида $k^{-1}([P])$ и $k^{-1}([c])$ обозначаются $k^{-1}[P]$ и $k^{-1}[c]$ соответственно. Выражения вида $\{e\}$, где $e \in Tm$, обозначаются без фигурных скобок.

Ниже каждому выражению сопоставляется **значение** этого выражения в текущий момент времени, которое является множеством термов.

2.1.6. Формулы

В этом пункте определяется понятие **формулы**, которое предназначено для описания свойств множеств термов. В определении данного понятия используется понятие **элементарной формулы (ЭФ)**, которая представляет собой запись одного из следущих видов:

1) $e \in E$, $E = E'$, $E \subseteq E'$, $E \supseteq E'$, где $e \in Tm$, $E, E' \in Expr$,
2) $E \perp_{\mathbf{C}} P$ и $E \perp_{\mathbf{K}} P$, где $E \subseteq Tm, P \in Processes$,
3) $at_P = i$, где $P \in Processes$.

ЭФ выражают свойства значений входящих в них выражений в текущий момент времени. ЭФ из первого пункта выражают свойства, соответствующие входящим в них теоретико-множественным символам. ЭФ из второго пункта выражают свойства, изложенные в пункте 2.2.4, ЭФ из третьего пункта выражают свойства текущего состояния последовательного процесса, подробнее см. в пункте 2.2.3.

Примеры ЭФ:

$$\left.\begin{array}{l} decrypt(k, k(e)) = e, \quad \text{где } k \in Var_{\mathbf{K}}, e \in Tm \\ pr_{n,i}(e_1, \ldots, e_n) = e_i, \quad \text{где } n > 0, i \in \{1, \ldots, n\}, e_1, \ldots, e_n \in Tm. \end{array}\right\} \quad (2)$$

Формулой называется произвольная совокупность ЭФ. Каждая формула $\varphi = \{\varphi_i \mid i \in I\}$ выражает утверждение, представляющее собой конъюнкцию утверждений, выражаемых ЭФ $\varphi_i \;\; (i \in I)$.

Множество всех формул обозначается записью Fm. $\forall\, \varphi \in Fm$ запись $Var(\varphi)$ обозначает множество всех переменных, входящих в φ.

Для каждого списка формул φ_1, …, $\varphi_n \in Fm$ формула $\varphi_1 \cup \ldots \cup \varphi_n$ будет обозначаться записью $\{\varphi_1, \ldots, \varphi_n\}$.

Формулы вида $\neg(e \in E)$ будут обозначаться записями вида $e \notin E$.

Запись вида $E_0 \rho_1 E_1 \ldots \rho_n E_n$, где $E_0, \ldots, E_n$ – выражения, и ρ_1, …, ρ_n – символы из $\{=, \subseteq\}$, обозначает формулу $\{E_0 \rho_1 E_1, \ldots, E_{n-1} \rho_n E_n\}$.

2.1.7. Связывания

Связыванием называется произвольная функция $\theta : Var \to Tm$. Будем говорить, что связывание θ связывает переменную $x \in Var$ с термом $\theta(x)$.

Будем использовать следующие обозначения:

- множество всех связываний обозначается символом Θ,
- id обозначает тождественное связывание: $\forall\, x \in Var \;\; id(x) = x$,
- $\forall\, X \subseteq Var \quad \Theta(X) = \{\theta \in \Theta \mid \forall\, x \in Var \setminus X \;\; \theta(x) = x\}$,
- связывание $\theta \in \Theta$ может обозначаться записями

$$x \mapsto \theta(x) \quad \text{или} \quad (\theta(x_1)/x_1, \ldots, \theta(x_n)/x_n), \tag{3}$$

  вторая запись в (3) используется, когда $\theta \in \Theta(\{x_1, \ldots, x_n\})$,
- $\forall\, \theta \in \Theta,\ \forall\, e \in Tm$ запись e^θ обозначает терм, получаемый из e заменой $\forall\, x \in Var(e)$ каждого вхождения x в e на терм $\theta(x)$, терм e называется **шаблоном** терма e^θ относительно связывания θ,
- $\forall\, \theta \in \Theta,\ \forall\, E \subseteq Tm$ запись E^θ обозначает множество $\{e^\theta \mid e \in E\}$,
- $\forall\, \theta, \theta' \in \Theta$ запись $\theta\theta'$ обозначает связывание $x \mapsto (x^\theta)^{\theta'}$.

Пусть $X \subseteq X' \subseteq Var$, $\theta \in \Theta(X)$, $\theta' \in \Theta(X')$. Связывание θ' называется **продолжением** связывания θ, если $\forall\, x \in X \;\; \theta(x) = \theta'(x)$.

2.2. Последовательные процессы

2.2.1. Действия

Действие – это запись одного из следующих видов:

$$c!e, \quad c?e, \quad e := e', \quad \text{где } c \in Channels,\ e, e' \in Tm,$$

которые называются **посылкой** сообщения e в канал c, **приемом** сообщения e из канала c, и **присваиванием**, соответственно.

Множество всех действий обозначается записью Act. $\forall\, \alpha \in Act$ множество всех переменных, входящих в α, обозначается записью $Var(\alpha)$.

Если $\theta \in \Theta$ и $\alpha \in Act$, то запись α^θ обозначает действие $c^\theta!e^\theta$, $c^\theta?e^\theta$ и $e^\theta := (e')^\theta$, если $\alpha = c!e$, $c?e$ и $e := e'$, соответственно.

В некоторых случаях, для облегчения визуального восприятия, действия могут записываться в круглых скобках, т.е., например, вместо записи $c!e$ может использоваться запись $(c!e)$, и т.д.

2.2.2. Понятие последовательного процесса

Последовательный процесс (ПП) – это четверка $(P, A, X, \bar{X})$, компоненты которой имеют следующий смысл:

- P – граф с выделенной вершиной (называемой **начальной вершиной**, и обозначаемой записью $Init(P)$), каждому ребру которого сопоставлена метка $\alpha \in Act$,
- A – **агент**, связанный с этим ПП,
- $X \subseteq Var$ – **инициализированные переменные**,
- $\bar{X} \subseteq X$ – **скрытые переменные**, они обозначают секретные ключи, скрытые каналы, или нонсы, эти переменные инициализированы уникальными значениями.

ПП является формальным описанием поведения динамической системы, работа которой заключается в последовательном выполнении действий, связанных с посылкой или приемом сообщений, а также с инициализацией неинициализированных переменных.

Для каждого ПП $(P, A, X, \bar{X})$

- данный ПП может сокращенно обозначаться тем же символом P, что и соответствующий ему граф, множество вершин графа P также обозначается символом P,
- начальная вершина P обозначается символом $\odot$; те вершины P, из которых не выходит ни одного ребра, обозначаются символом $\otimes$,
- $Agent(P)$, $X(P)$, $\bar{X}(P)$ обозначают соответствующие компоненты P, $Var(P)$ обозначает множество всех переменных, входящих в P,
- $\tilde{X}(P)$ обозначает множество $X(P) \setminus \bar{X}(P)$ инициализированных нескрытых переменных процесса P,
- $\hat{X}(P)$ обозначает множество $Var(P) \setminus X(P)$ неинициализированных переменных процесса P.

С каждым ПП связана переменная из множества $Processes$, называемая **именем** этого ПП. Будем обозначать имена ПП теми же записями, которыми обозначаются сами ПП.

Если P не имеет ребер и $X(P) = \emptyset$, то P обозначается символом $\mathbf{0}$.

Действия вида $\circ!e$ и $\circ?e$ будут более коротко обозначаться записями $!e$ и $?e$ соответственно.

2.2.3. Состояние последовательного процесса

Состояние ПП P – это пятерка $s = (at, \alpha, [P], \theta, \{[c] \mid c \in \mathit{Channels}\})$, где

- $at \in P$ – вершина графа P в состоянии s,
- $\alpha \in \{\mathit{init}\} \sqcup \mathit{Act}$ – действие перед переходом в s,
- $[P] \subseteq \mathit{Var}$ – множество инициализированных переменных в s,
- $\theta \in \Theta([P])$ – связывание в s,
- $\forall\, c \in \mathit{Channels}\ \ [c] \subseteq Tm$ – содержимое канала c в s.

Компоненты состояния s обозначаются записями at_s, α_s, $[P]_s$, θ_s, $[c]_s$ соответственно. Будем обозначать записью $\langle P \rangle_s$ множество $Tm([P]_s)$.

Состояние ПП P называется **начальным** (и обозначается 0_P), если оно имеет вид $(\mathit{Init}(P), \mathit{init}, X(P), \mathit{id}, \{\emptyset \mid c \in \mathit{Channels}\})$.

2.2.4. Значения выражений и формул в состояниях последовательных процессов

Пусть заданы ПП P, состояние s ПП P, выражение E, и формула φ.

Запись E^s обозначает множество термов, называемое **значением** E в s, и определяемое следующим образом:

- $\forall\, E \subseteq Tm \quad E^s = \{e^{\theta_s} \mid e \in E\}$,
  $\forall\, e \in Tm$ множество вида $\{e\}^s$, а также единственный элемент этого множества, будем обозначать записью e^s,
- $[P]^s = ([P]_s)^s$, $\langle P \rangle^s = (\langle P \rangle_s)^s$, $[c]^s = [c^s]_s$, где $P \in \mathit{Processes}$, $c \in \mathit{Channels}$,
- $k^{-1}(E)^s = \{e \in Tm \mid \exists\, e' \in E^s : k^s(e) \subseteq e'\}$,
- $(E \cap E')^s = E^s \cap (E')^s$, $(E \cup E')^s = E^s \cup (E')^s$, $(\neg E)^s = Tm \setminus E^s$.

Запись $s \models \varphi$ обозначает утверждение "φ **истинна в** s". Это утверждение верно, если $\mathit{Var}(\varphi)_{\mathbf{P}} \subseteq \{P\}$, и выполнено одно из условий:

- $\varphi = (e \in E)$, $(E = E')$, $(E \subseteq E')$, или $(E \supseteq E')$, где $e \in Tm$, $E, E' \in \mathit{Expr}$, и

  $$e^s \in E^s,\ E^s = (E')^s,\ E^s \subseteq (E')^s,\ E^s \supseteq (E')^s,\ \text{соответственно}$$

- $\varphi = (E \perp_{\mathbf{C}} P)$, $\forall\, e \in E^s \;\; Agent(P) \notin e$, и

$$\left.\begin{array}{l} \forall\, x \in E^s_{\mathbf{X}}, \forall\, y \in [P]_s \;\; x \notin y^s \\ \forall\, x \in E^s_{\mathbf{X}}, \forall\, c \in Channels \;\; \text{если } \exists\, e \in [c]_s : x \in e, \;\text{то } c \in E^s \end{array}\right\} \quad (4)$$

(4) можно интерпретировать как следующее утверждение: каждая переменная из $E^s_{\mathbf{X}}$ не входит в термы, доступные процессу P в состоянии s, и входит в термы из содержимого только таких каналов, которые недоступны для P,

- $\varphi = (E \perp_{\mathbf{K}} P)$, $\forall\, e \in E^s \;\; Agent(P) \notin e$, и

$$\left.\begin{array}{l} \forall\, x \in E^s_{\mathbf{X}},\, \forall\, y \in [P]_s \;\; x \perp_{\mathbf{K},E} y^s \\ \forall\, x \in E^s_{\mathbf{X}},\, \forall\, c \in Channels, \forall\, e \in [c]_s \;\; x \perp_{\mathbf{K},E} e \end{array}\right\} \quad (5)$$

где $x \perp_{\mathbf{K},E} e$, означает, что

$$\begin{array}{l} \text{каждое вхождение } x \text{ в } e \text{ содержится} \\ \text{в подтерме } k(\ldots) \subseteq e, \text{ где } k \in E^s_{\mathbf{K}} \end{array} \quad (6)$$

(5) можно интерпретировать как следующее утверждение: переменные из $E^s_{\mathbf{X}}$ входят в термы, доступные процессу P в состоянии s, а также в термы из содержимого произвольного канала, в "защищённом" виде, т.е. входят в подтермы вида $k(\ldots)$, где $k \in E^s_{\mathbf{K}}$,

- $\varphi = (at_P = i)$, и $at_s = i$,
- $\varphi = \{\varphi_i \mid i \in I\}$ – совокупность ЭФ, и $\forall\, i \in I \;\; s \models \varphi_i$.

2.2.5. Выполнение последовательного процесса

Выполнение ПП P можно понимать как обход вершин P, начиная с $Init(P)$, с выполнением действий, являющихся метками проходимых рёбер. С каждым шагом выполнения ПП P связано некоторое состояние s ПП P, называемое **текущим состоянием** ПП P на этом шаге (на первом шаге текущим состоянием является 0_P).

Если текущий шаг выполнения ПП P не является заключительным, то на этом шаге происходит замена текущего состояния s на состояние s', которое будет текущим состоянием на следующем шаге, для этого

1) либо выбирается выходящее из at_s ребро графа P, метка α которого обладает следующими свойствами:

   - если α^{θ_s} содержит вхождение терма вида $shared\_key(\ldots)$, или $shared\_channel(\ldots)$, то $Agent(P)$ присутствует в этом вхождении,

- выполнено одно из условий:

$$\left.\begin{array}{ll}
\text{(a)} & \alpha = c!e,\ c, e \in \langle P\rangle_s \\
\text{(b)} & \alpha = c?e,\ c \in \langle P\rangle_s, \\
 & \quad VarEncKeys(e^s) \subseteq [P]_s, \\
 & \quad OpenEncKeys(e^s) \subseteq \{Agent(P)\}, \\
 & \quad \exists\, \theta \in \Theta(\, Var(e) \setminus [P]_s) : (e^\theta)^s \in [c]^s \\
\text{(c)} & \alpha = (e := e'),\ e' \in \langle P\rangle_s, \\
 & \quad VarEncKeys(e^s) \subseteq [P]_s, \\
 & \quad OpenEncKeys(e^s) \subseteq \{Agent(P)\}, \\
 & \quad \exists\, \theta \in \Theta(\, Var(e) \setminus [P]_s) : e^\theta = e'
\end{array}\right\} \qquad (7)$$

и компоненты состояния s' имеют следующий вид: $at_{s'}$ – конец выбранного ребра, $\alpha_{s'} = \alpha$, и

- если верно (a) в (7), то $[P]_{s'} = [P]_s$, $\theta_{s'} = \theta_s$, $[c^s]_{s'} = [c^s]_s \cup \{e^s\}$, $\forall\, c' \in Channels \setminus \{c^s\}\ \ [c']_{s'} = [c']_s$,
- если верно (b) или (c) в (7), то $[P]_{s'} = [P]_s \cup Var(e)$, $\theta_{s'} = \theta\theta_s$, $\forall\, c' \in Channels\ \ [c']_{s'} = [c']_s$,
  (будем говорить, что при переходе от s к s' каждая переменная $x \in Var(e) \setminus [P]_s$ инициализируется значением $x^{\theta_{s'}}$, которое становится доступным P),

2) либо все компоненты состояния s', кроме последней, совпадают с соответствующими компонентами состояния s, и $\forall\, c \in Channels$ множество $[c]_{s'}$ либо совпадает с $[c]_s$, либо получается путем добавления терма к множеству $[c]_s$ в результате выполнения текущего шага другим ПП.

Если имеет место первая (вторая) из указанных выше ситуаций, то будем говорить, что s' получается **активным** (соответственно, **пассивным**) переходом из s. Запись $s \overset{P}{\rightarrow} s'$ ($s \rightarrow s'$) обозначает, что s' получается активным (соответственно, пассивным) переходом из s.

Во время каждого выполнения каждого ПП P переменные из $Var(P)$ имеют следующие особенности: $\forall\, x \in Var(P)$

1) если $x \in \hat{X}(P)$, то в начальный момент каждого выполнения ПП P переменная x не инициализирована, т.е. ей не сопоставлено никакого значения,

2) если $x \in \bar{X}(P)$, то это означает, что в начальный момент каждого выполнения $Exec$ ПП P данная переменная инициализирована **уникальным значением**, т.е. значением, которое отличается

- от значений, сопоставленных другим инициализированным переменным при выполнении $Exec$, и
- от значений, сопоставленных инициализированным переменным при любом выполнении $Exec' \neq Exec$ любого ПП.

Интерпретация условий, описанных в (7), имеет следующий вид.

- Условие в пункте (a) связано с выполнением посылки сообщения:
  - имя c^s канала, в который посылается сообщение, должно быть доступно процессу P в состоянии s, и
  - посылаемое сообщение e^s должно быть термом, компоненты которого также доступны процессу P в состоянии s.
- Условие в пункте (b) связано с выполнением приёма сообщения:
  - имя c^s канала, из которого принимается сообщение, должно быть доступно процессу P в состоянии s,
  - все ШС в принимаемом сообщении, которые
    - дешифруются во время приёма этого сообщения, и
    - зашифрованы не на открытом или разделяемом ключе,

    имеют вид $k(\ldots)$, где значение ключа k должно быть доступно процессу P в состоянии s, это свойство выражается во второй строке пункта (b) в (7),
  - все ШС в принимаемом сообщении, которые
    - дешифруются во время приёма этого сообщения, и
    - зашифрованы на открытом ключе,

    имеют вид $Agent(P)(e)$, это свойство выражается в третьей строке пункта (b) в (7),
  - терм e является шаблоном некоторого терма из $[c]^s$ относительно некоторого продолжения связывания θ_s, данное свойство выражается в последней строке пункта (b) в (7).
- Условие в пункте (c) связано с выполнением присваивания:
  - каждая компонента терма $(e')^s$ должна быть доступна P в s,
  - смысл свойств во второй и третьей строках пункта (c) в (7) совпадает со смыслом соответствующих свойств в пункте (b): каждое ШС в $(e')^s$, которое должно быть дешифровано во время выполнения этого присваивания, должно иметь вид $k(\ldots)$ или $Agent(P)(\ldots)$, причем

  - либо k разделяемый ключ,
  - либо $k \in Var_{\mathbf{K}}$ и значение ключа k должно быть доступно P в состоянии s,
- терм e является шаблоном терма e' относительно некоторого связывания $\theta \in \Theta(\, Var(e) \setminus [P]_s)$.

2.2.6. Процесс противника

Процесс противника – это ПП, обозначаемый записью $P_\dagger$, и обладающий следующими свойствами:

- граф ПП $P_\dagger$ состоит из единственной вершины,
- $\forall\, \tau \in Types$ множества $\bar{X}(P_\dagger)_\tau$ и $\hat{X}(P_\dagger)_\tau$ счетны,
- $\forall\, \alpha \in Act$ граф $P_\dagger$ содержит ребро с меткой α.

Ниже будем предполагать, что $P_\dagger$ – единственный из всех рассматриваемых ПП, граф которого имеет циклы.

2.2.7. Переименование переменных

Переименование переменных (называемое также просто **переименованием**) – это инъективная функция $\eta : X \to X'$, где $X, X' \subseteq Var$.

Для каждого переименования $\eta : X \to X'$, каждого $e \in Tm$ и каждого ПП P записи e^η и P^η обозначают терм или ПП соответственно, получаемые из e или P заменой $\forall\, x \in X$ каждого вхождения x на $\eta(x)$.

Если переименование η имеет вид $\eta : \bar{X}(P) \cup \hat{X}(P) \to Var \setminus \tilde{X}(P)$, то ПП P и P^η будем рассматривать как равные.

2.3. Распределенные процессы

В этом пункте вводится понятие распределенного процесса, которое является моделью КП. Все КП, рассматриваемые в этом тексте, мы будем отождествлять с соответствующими им распределенными процессами.

2.3.1. Понятие распределенного процесса

Распределенный процесс (РП) – это семейство ПП:

$$\mathcal{P} = \{P_i \mid i \in I\}$$

(некоторые из которых могут совпадать). С каждым РП связана переменная типа $\mathbf{P}$, называемая **именем** этого РП.

РП $\mathcal{P}$ является моделью распределенного алгоритма, компонентами которого являются входящие в него ПП, взаимодействующие друг с другом путем передачи сообщений через каналы.

Пусть задан РП $\mathcal{P}$. Будем использовать следующие обозначения и предположения:

- $Var(\mathcal{P}) = \bigcup_{P \in \mathcal{P}} Var(P)$, множества $X(\mathcal{P})$, $\bar{X}(\mathcal{P})$, $\tilde{X}(\mathcal{P})$, $\hat{X}(\mathcal{P})$ определяются аналогично,

- будем предполагать, что

$$\begin{array}{l}\text{компоненты семейства } \{\bar{X}(P) \cup \hat{X}(P) \mid P \in \mathcal{P}\} \\ \text{дизъюнктны и не пересекаются с } \tilde{X}(\mathcal{P})\end{array} \tag{8}$$

  (если это не так, то заменим каждый из компонентов P семейства $\mathcal{P}$ на равный ему в том смысле, который указан в конце пункта 2.2.7, так, чтобы свойство (8) выполнялось),

- РП $\mathcal{P}$ может обозначаться записью
  - $\{P_1, \ldots, P_n\}$, если $I = \{1, \ldots, n\}$ (в случае $n = 1$ скобки м.б. опущены, т.е. вместо $\{P_1\}$ пишется P_1), или
  - P^*, если I – множество натуральных чисел, и все ПП, входящие в $\mathcal{P}$, совпадают с P,

- запись $\mathcal{P}_\dagger$ обозначает РП $\{\mathcal{P}, P_\dagger\}$,

- если $\{\mathcal{P}_i \mid i \in I\}$ – семейство РП, и $\forall\, i \in I$ РП $\mathcal{P}_i$ является семейством ПП вида $\{P_{i'} \mid i' \in I_i\}$, где множества индексов I_i $(i \in I)$ дизъюнктны (если это не так, то заменим их на соответствующие дизъюнктные копии), то запись $\{\mathcal{P}_i \mid i \in I\}$ обозначает также РП $\{P_{i'} \mid i' \in \bigsqcup_{i \in I} I_i\}$.

Будем использовать следующее соглашение:

- если в каком-либо рассуждении, связанном с РП вида P^*, некоторый ПП является первым из рассматриваемых ПП, входящих в P^*, то этот ПП и все его переменные обозначаются теми же записями, которые используются в ПП P,

- если кроме этого ПП рассматривается другой ПП, входящий в P^*, то он уже обозначается записью $\grave{P}$, и в обозначениях тех его переменных, которые соответствуют переменным из $\bar{X}(P) \cup \hat{X}(P)$, используются обратные штрихи, и т.д.

2.3.2. Понятие состояния распределенного процесса

Пусть задан РП $\mathcal{P}$.

Состоянием РП $\mathcal{P}$ называется семейство $s = \{s_P \mid P \in \mathcal{P}\}$ состояний ПП, входящих в $\mathcal{P}$, такое, что $\forall\, c \in \mathit{Channels}$ все множества в семействе $\{[c]_{s_P} \mid P \in \mathcal{P}\}$ одинаковы (будем обозначать их записью $[c]_s$).

Пусть $s = \{s_P \mid P \in \mathcal{P}\}$ – состояние РП $\mathcal{P}$. Тогда

- s называется **начальным** состоянием РП $\mathcal{P}$, и обозначается $0_{\mathcal{P}}$, если $\forall\, P \in \mathcal{P}\ \ s_P = 0_P$,
- $at_s = \{at_{s_P} \mid P \in \mathcal{P}\}$, $[\mathcal{P}]_s = \bigcup_{P \in \mathcal{P}} [P]_s$, $\langle \mathcal{P} \rangle_s = Tm([\mathcal{P}]_s)$,
- θ_s обозначает связывание из $\Theta([\mathcal{P}]_s)$, такое, что

$$\forall\, P \in \mathcal{P},\ \forall\, x \in [P]_s \quad \theta_{s_P}(x) = \theta_s(x)$$

  (существование такого связывания следует из предположения (8)).

Понятия значения выражения и значения формулы в состоянии РП определяются аналогично соответствующим понятиям для ПП.

$\forall\, \varphi, \psi \in Fm$ запись $\varphi \leq \psi$ означает, что для каждого РП $\mathcal{P}$ и каждого состояния s РП $\mathcal{P}$ верна импликация $s \models \varphi \Rightarrow s \models \psi$.

Если формулы $\varphi, \psi \in Fm$ таковы, что $\varphi \leq \psi$ и $\psi \leq \varphi$, то будем рассматривать такие формулы как одинаковые. Если формулы φ и ψ одинаковы, то будем обозначать этот факт записью $\varphi = \psi$.

Примеры одинаковых формул:

- $\{f(e_1, \ldots, e_n) = f(e'_1, \ldots, e'_n)\}$, где $f \in Fun$, и $\{e_1 = e'_1, \ldots, e_n = e'_n\}$,
- $\{[c] = \{e\}, e' \in [c]\}$ и $\{[c] = \{e\}, e = e'\}$.

2.3.3. Выполнение распределённого процесса

Пусть задан РП $\mathcal{P}$. **Выполнение** РП $\mathcal{P}$ представляет собой недетерминированное чередование выполнений ПП, входящих в $\mathcal{P}$. На каждом шаге выполнения РП $\mathcal{P}$

- только один ПП из $\mathcal{P}$ выполняет активный переход, и
- остальные ПП из $\mathcal{P}$ выполняют пассивные переходы.

Выполнение РП $\mathcal{P}$ можно определить как порождение последовательности состояний этого РП (начиная с начального состояния $0_{\mathcal{P}}$), в которой каждое состояние s, не являющееся последним в этой последовательности, связано со следующим состоянием s' **отношением перехода**, что

означает следующее: $\exists\, P \in \mathcal{P}$:

$$\begin{array}{l} \overset{P}{s_P} \to\ s'_P,\ \ \forall\, P' \in \mathcal{P} \setminus \{P\}\ \ s_{P'} \to s'_{P'} \\ \text{где } s = \{s_P \mid P \in \mathcal{P}\},\ s' = \{s'_P \mid P \in \mathcal{P}\}. \end{array} \tag{9}$$

Свойство (9) обозначается записью $s \xrightarrow{\alpha_P} s'$, где $\alpha = \alpha_{s'_P}$.

Множество всех состояний РП $\mathcal{P}$ можно рассматривать как граф, в котором существует ребро из s в s' с меткой α_P тогда и только тогда когда $s \xrightarrow{\alpha_P} s'$. Обозначение ПП P в метке α_P можно опускать.

Для каждой пары состояний s, s' РП $\mathcal{P}$ запись $s \to s'$ означает, что s связано с s' отношением перехода, и запись $s \Rightarrow s'$ означает, что существует последовательность $s_1, \ldots, s_n$ состояний, такая, что $s_1 = s$, $s_n = s'$, и $\forall\, i = 1, \ldots, n-1\ \ s_i \to s_{i+1}$.

Состояние s РП $\mathcal{P}$ называется **достижимым**, если $0_{\mathcal{P}} \Rightarrow s$. Множество достижимых состояний РП $\mathcal{P}$ обозначается записью $\Sigma_{\mathcal{P}}$.

Если задан путь π из $0_{\mathcal{P}}$ в s, и s' – какое-либо состояние, входящее в π, то мы будем обозначать этот факт записью $s' \leq_\pi s$. Запись $s' <_\pi s$ обозначает, что $s' \leq_\pi s$ и $s' \neq s$. Если путь π ясен из контекста, то обозначение этого пути в записях $\leq_\pi$ и $<_\pi$ м.б. опущено.

2.3.4. Наблюдаемая эквивалентность

Понятие наблюдаемой эквивалентности РП имеет следующий нефомальный смысл: РП $\mathcal{P}$ и $\mathcal{P}'$ наблюдаемо эквивалентны, если для каждого наблюдателя, который анализирует выполнение РП $\mathcal{P}_\dagger$ и $\mathcal{P}'_\dagger$ путем наблюдения за содержимым канала $\circ$, эти РП будут неразличимы.

Пусть $\mathcal{P}$, $\mathcal{P}'$ – РП, $s \in \Sigma_{\mathcal{P}_\dagger}$, $s' \in \Sigma_{\mathcal{P}'_\dagger}$, и $\eta : X \to X'$ – переименование. Запись $s \underset{\eta}{\sim} s'$ означает, что $[\circ]_s \subseteq Tm(X)$ и $[\circ]_{s'} = \{e^\eta \mid e \in [\circ]_s\}$.

Будем говорить, что РП $\mathcal{P}$ и $\mathcal{P}'$ **наблюдаемо эквивалентны**, если существует множество μ троек вида (s, s', η), где $s \in \Sigma_{\mathcal{P}_\dagger}$, $s' \in \Sigma_{\mathcal{P}'_\dagger}$, $s \underset{\eta}{\sim} s'$, такое, что выполнены следующие условия:

- $(0_{\mathcal{P}_\dagger}, 0_{\mathcal{P}'_\dagger}, \emptyset) \in \mu$ (где $\emptyset$ - функция с пустой областью определения),
- $\forall\, (s, s', \eta) \in \mu$ если $s \to \tilde{s}$ или $s' \to \tilde{s}'$, то $\exists\, (\tilde{s}, \tilde{s}', \tilde{\eta}) \in \mu$: $\tilde{\eta}$ – продолжение η, и $s' \Rightarrow \tilde{s}'$ или $s \Rightarrow \tilde{s}$, соответственно.

Отметим, что данное определение наблюдаемой эквивалентности не является единственно возможным, и м.б. модифицировано в зависимости от решаемой задачи. В некоторых задачах более подходящим определением наблюдаемой эквивалентности является такое огрубление опреде-

ленного выше отношения, относительного которого будут эквивалентными РП $\mathcal{P} = \{P\}$, $\mathcal{P}' = \{P'\}$, где

$$P = \odot \xrightarrow{!k(e)} \otimes,\ P' = \odot \xrightarrow{!k'(e')} \otimes,\ \ k \in \bar{X}(P)_{\mathbf{K}},\ k' \in \bar{X}(P')_{\mathbf{K}}.$$

2.4. Теоремы о сохранении значений формул при переходах распределенных процессов

В этом параграфе формулируются и доказываются теоремы о сохранении значений некоторых формул при переходах РП. Данные теоремы используются при решении задач верификации РП. В излагаемых ниже примерах применения этих теорем

- ПП P, упоминаемый в этих теоремах – это ПП противника $P_\dagger$, и
- говоря неформально, данные теоремы утверждают что
  - если имена каких-либо каналов защищены от противника, то содержимое этих каналов не м.б. изменено противником, и
  - если какие-либо криптографические ключи защищены от противника, то содержимое ШС, зашифрованных на этих ключах, недоступно противнику.

2.4.1. Теоремы о защищенных каналах

Первая теорема связана с сохранением значений формул вида

$$E \perp_{\mathbf{C}} P, \quad \text{где } E \subseteq Tm, \text{ и } P - \text{ПП} \tag{10}$$

при переходах между состояниями. Данная теорема м.б. интерпретирована как следующее утверждение: если

- какому-либо из ПП, входящих в некоторый РП, недоступны сообщения, выражаемые термами из некоторого множества E, и
- в текущем состоянии этого РП ни в каком из каналов, доступных этому ПП, нет сообщений, соответствующих термам из E,

то никакая собственная активность этого ПП не приведет к тому, что сообщения, выражаемых термами из E, станут доступны этому ПП.

Каналы, имена которых входят в E, могут интерпретироваться как **защищённые** каналы относительно действий ПП P.

Теорема 1.
Пусть РП $\mathcal{P}$, ПП $P \in \mathcal{P}$, и состояния $s, s' \in \Sigma_{\mathcal{P}}$, таковы, что $s \xrightarrow{\alpha_P} s'$.

Тогда $\forall\, E \subseteq \langle \mathcal{P} \rangle_0$ верна импликация

$$s \models E \perp_{\mathbf{C}} P \;\; \Rightarrow \;\; s' \models E \perp_{\mathbf{C}} P.$$

Доказательство.
Напомним, что $s \models E \perp_{\mathbf{C}} P$ означает, что $\forall\, e \in E \;\; Agent(P) \notin e$, и

$$\left.\begin{array}{l} \forall\, x \in E_{\mathbf{X}}, \forall\, y \in [P]_s \;\; x \notin y^s \\ \forall\, x \in E_{\mathbf{X}}, \forall\, c \in Channels \;\; \text{если } \exists\, e \in [c]_s : x \in e, \text{ то } c \in E \end{array}\right\} \qquad (11)$$

Докажем, что из (11) следует $s' \models E \perp_{\mathbf{C}} P$, т.е.

$$\left.\begin{array}{l} \forall\, x \in E_{\mathbf{X}}, \forall\, y \in [P]_{s'} \;\; x \notin y^{s'} \\ \forall\, x \in E_{\mathbf{X}}, \forall\, c \in Channels \;\; \text{если } \exists\, e \in [c]_{s'} : x \in e, \text{ то } c \in E \end{array}\right\} \qquad (12)$$

1) Пусть неверно первое утверждение в (12). Тогда из первого утверждения в (11) следует, что $[P]_s \neq [P]_{s'}$, и имеет место один из двух перечисленных ниже случаев.

   - Первый случай:

     $$\left.\begin{array}{l} \alpha = c?e, \text{ где } c \in \langle P \rangle_s, e^{s'} \in [c^s]_s \\ [P]_{s'} = [P]_s \cup Var(e), \exists\, x \in E_{\mathbf{X}}, \exists\, y \in Var(e) : x \in y^{s'}. \end{array}\right\} \qquad (13)$$

     Из соотношений $x \in y^{s'} \subseteq e^{s'} \in [c^s]_s$ и второго утверждения в (11) следует, что $c^s \in E$.

     Если $c^s \notin E_{\mathbf{X}}$, то c имеет вид $shared\_channel(\ldots)$, и в этом случае из определения понятия выполнения ПП в пункте 2.2.5 следует, что $Agent(P) \in c^s$. Однако это и утверждение $c^s \in E$ противоречат предположению $\forall\, e \in E \;\; Agent(P) \notin e$.

     Таким образом, $c^s \in E_{\mathbf{X}}$. Отсюда следует, что $c \in [P]_s$.

     Согласно первому утверждению в (11) (в котором в качестве x и y берем c^s и c соответственно), должно быть верно утверждение $c^s \notin c^s$, которое является ложным.

   - Второй случай:

     $$\left.\begin{array}{l} \alpha \text{ имеет вид } e := e', \text{ где } e' \in \langle P \rangle_s, e^{s'} = (e')^s, \\ [P]_{s'} = [P]_s \cup Var(e), \exists\, x \in E_{\mathbf{X}}, \exists\, y \in Var(e) : x \in y^{s'} \end{array}\right\} \qquad (14)$$

     Из соотношений $x \in y^{s'} \subseteq e^{s'} = (e')^s \in \langle P \rangle^s$ следует, что

     $$\exists\, z \in [P]_s : x \in z^s. \qquad (15)$$

     Однако (15) противоречит первому утверждению в (11).

2) Пусть неверно второе утверждение в (12), т.е.

$$\exists\, x \in E_{\mathbf{X}}, \exists\, c' \in Channels, \exists\, e' \in [c']_{s'} : x \in e',\ \text{но}\ c' \notin E.$$

Тогда из второго утверждения в (11) следует, что $[c']_{s'} \neq [c']_s$, и

$$\alpha \text{ имеет вид } c!e, \text{ где } c, e \in \langle P \rangle_s, x \in e' = e^s.$$

Из $x \in e^s$ следует, что $\exists\, y \in [P]_s : x \in y^s$, что противоречит первому утверждению в (11). ■

Следующая теорема является усилением теоремы 1. В ней утверждается, что в условиях теоремы 1 нижняя и верхняя границы на содержимое защищенных каналов не изменяются при выполнении действий P.

Теорема 2.
Пусть РП $\mathcal{P}$, ПП $P \in \mathcal{P}$, и состояния $s, s' \in \Sigma_{\mathcal{P}}$, таковы, что $s \xrightarrow{\alpha_P} s'$. Тогда $\forall\, E \subseteq \langle \mathcal{P} \rangle_0$, $\forall\, E', E'' \subseteq Tm$, $\forall\, c \in E_{\mathbf{C}}$ верна импликация

$$s \models \varphi \Rightarrow s' \models \varphi, \quad \text{где } \varphi = \{E \perp_{\mathbf{C}} P, E' \subseteq [c] \subseteq E''\}.$$

Доказательство.
Согласно теореме 1, из $s \models E \perp_{\mathbf{C}} P$ следует $s' \models E \perp_{\mathbf{C}} P$. Кроме того, $[c]_s \subseteq [c]_{s'}$. Таким образом, для доказательства теоремы достаточно доказать импликацию

$$s \models \varphi \quad \Rightarrow \quad s' \models [c] \subseteq E''. \tag{16}$$

Если заключение импликации (16) неверно, то $[c]_s \neq [c]_{s'}$.

Т.к. по предположению $c \in E_{\mathbf{C}} \subseteq \langle \mathcal{P} \rangle_0$, то неравенство $[c]_s \neq [c]_{s'}$ возможно только если α имеет вид $c'!e$, где $(c')^s = c$.

Если c не является переменной, то c является разделяемым каналом, и по определению выполнения действия вида $c'!e$, в этом случае должно быть выполнено условие $Agent(P) \in c$, что противоречит предположению $s \models E \perp_{\mathbf{C}} P$ (т.к. в частности должно быть верно, что $Agent(P)$ не входит в термы из E). Следовательно, $c \in Var$, поэтому $c' \in Var$, и $c' \in [P]_s$.

Т.к. $c \in E_{\mathbf{X}}$ и $c' \in [P]_s$, то согласно предположению $s \models E \perp_{\mathbf{C}} P$, должно быть верно утверждение $c \notin c$, которое является ложным. ■

2.4.2. Теоремы о защищённых ключах

В этом пункте доказываются теоремы, аналогичные теоремам 1 и 2. В них вместо защищенных каналов рассматриваются защищенные ключи.

Первая теорема связана с сохранением значений формул вида

$$E \perp_{\mathbf{K}} P, \quad \text{где } E \subseteq Tm, \text{ и } P - \text{ПП} \tag{17}$$

при переходах между состояниями. Данная теорема м.б. интерпретирована как следующее утверждение: если

- какому-либо из ПП, входящих в некоторый РП, недоступны сообщения, выражаемые термами из некоторого множества E, и
- в текущем состоянии этого РП выполнено условие, выражаемое формулой (17),

то никакая собственная активность этого ПП не приведет к тому, что ключи, имена которых входят в E, когда-нибудь станут доступны этому ПП. Данные ключи могут интерпретироваться как **защищённые** ключи относительно действий ПП P.

Теорема 3.
Пусть РП $\mathcal{P}$, ПП $P \in \mathcal{P}$, и состояния $s, s' \in \Sigma_{\mathcal{P}}$, таковы, что $s \xrightarrow{\alpha_P} s'$. Тогда $\forall\, E \subseteq \langle \mathcal{P} \rangle_0$ верна импликация

$$s \models E \perp_{\mathbf{K}} P \Rightarrow s' \models E \perp_{\mathbf{K}} P.$$

Доказательство.
Напомним, что $s \models E \perp_{\mathbf{K}} P$ означает, что $\forall\, e \in E \;\; Agent(P) \notin e$, и

$$\left.\begin{array}{l} \forall\, x \in E_{\mathbf{X}}, \forall\, y \in [P]_s \;\; x \perp_{\mathbf{K},E} y^s \\ \forall\, x \in E_{\mathbf{X}}, \forall\, c \in Channels, \forall\, e \in [c]_s \;\; x \perp_{\mathbf{K},E} e \end{array}\right\} \tag{18}$$

Докажем, что из (18) следует $s' \models E \perp_{\mathbf{K}} e$, т.е. $\forall\, x \in E_{\mathbf{X}}$

$$\left.\begin{array}{l} \forall\, y \in [P]_{s'} \;\; s' \models x \perp_{\mathbf{K},E} y^{s'} \\ \forall\, c \in Channels, \forall\, e \in [c]_{s'} \;\; s' \models x \perp_{\mathbf{K},E} e \end{array}\right\} \tag{19}$$

1) Если первое утверждение в (19) неверно, то из первого утверждения в (18) следует, что $[P]_s \neq [P]_{s'}$, и имеет место один из двух случаев:

   - Первый случай: верно утверждение

$$\left.\begin{array}{l} \alpha \text{ имеет вид } c?e, c \in \langle P \rangle_s, e^{s'} \in [c^s]_s, \\ [P]_{s'} = [P]_s \cup Var(e), \exists\, y \in Var(e) : \\ \exists \text{ вхождение } x \text{ в } y^{s'}, \text{ которое не содержится} \\ \text{ни в каком подтерме вида } k(\ldots) \subseteq y^{s'}, \text{ где } k \in E_{\mathbf{K}}. \end{array}\right\} \tag{20}$$

Поскольку упомянутое в (20) вхождение x содержится в терме $y^{s'} \subseteq e^{s'} \in [c^s]_s$, то из второго утверждения в (18) следует, что это вхождение x содержится в подтерме $k(\tilde{e}) \subseteq e^{s'}$, где $k \in E_{\mathbf{K}}$. Из (20) следует, что $k(\tilde{e})$ не м.б. подтермом терма $y^{s'}$. Поскольку термы $k(\tilde{e})$ и $y^{s'}$ имеют непустое пересечение (оба содержат вышеупомянутое вхождение x), то из (1) следует, что $y^{s'} \subset k(\tilde{e})$. Таким образом,

$$y^{s'} \subset k(\tilde{e}) \subseteq e^{s'}. \tag{21}$$

Докажем индукцией по структуре терма e, что из (21) следует утверждение

$$\exists\, z \in Var(e) : \quad k(\tilde{e}) \subseteq z^{s'} \subseteq e^{s'}. \tag{22}$$

Если $e \in Con \cup Var$, то утверждение (22) верно.

Если e имеет вид $f(e_1, \ldots, e_n)$, где $f \in Fun$, то

- если $f = encrypt$, т.е. e имеет вид $k_1(e_1)$, то $k_1 \in Keys(e)$, согласно (7)(b) имеется включение $Keys(e) \subseteq [P]_s$, откуда следует, что $k_1 \in [P]_s$, и возможны следующие случаи:
  - $k(\tilde{e}) = e^{s'} = k_1^{s'}(e_1^{s'})$, в этом случае $k = k_1^{s'} = k_1^s \in [P]^s$, однако поскольку $k \in E_{\mathbf{K}}$, то по первому утверждению в (18) вхождение k в k должно содержаться в подтерме вида $k'(\ldots) \subseteq k$, что невозможно,
  - $k(\tilde{e}) \subseteq k_1^{s'}$, данный случай невозможен по определению термов типа $\mathbf{K}$,
  - $k(\tilde{e}) \subseteq e_1^{s'}$, в данном случае утверждение (22) следует из индуктивного предположения,
- если $f \neq encrypt$, то $\exists\, i \in \{1, \ldots, n\} : k(\tilde{e}) \subseteq e_i^{s'}$, и утверждение (22) следует из индуктивного предположения.

Из (21) и (22) следует, что

$$y^{s'} \subset k(\tilde{e}) \subseteq z^{s'} \subseteq e^{s'}. \tag{23}$$

Таким образом, терм e содержит вхождения переменных y и z, обладающие следующим свойством: $y^{s'} \subset z^{s'}$, откуда для данных вхождений следует включение $y \subset z$, что невозможно.

- Второй случай: верно утверждение

$$\left.\begin{array}{l} \alpha \text{ имеет вид } e := e', \text{ где } e' \in \langle P \rangle_s, e^{s'} = (e')^s, \\ [P]_{s'} = [P]_s \cup Var(e), \exists\, y \in Var(e) : \\ \exists \text{ вхождение } x \text{ в } y^{s'}, \text{ которое не содержится} \\ \text{ни в каком подтерме вида } k(\ldots) \subseteq y^{s'}, \text{ где } k \in E_{\mathbf{K}}. \end{array}\right\} \tag{24}$$

Поскольку упомянутое в (24) вхождение x содержится в терме $y^{s'} \subseteq e^{s'} = (e')^s$, то это вхождение x содержится в подтерме $(z')^s \subseteq (e')^s$, где $z' \in Var(e')$.

По предположению, $e' \in \langle P \rangle_s$, поэтому $Var(e') \subseteq [P]_s$, и, следовательно, $z' \in [P]_s$. Из первого утверждения в (18) следует, что упомянутое в (24) вхождение x в $(z')^s$ содержится в некотором подтерме $k(\tilde{e}) \subseteq (z')^s$, где $k \in E_{\mathbf{K}}$.

Из (24) следует, что $k(\tilde{e})$ не м.б. подтермом терма $y^{s'}$.

Поскольку термы $k(\tilde{e})$ и $y^{s'}$ имеют непустое пересечение (оба содержат упомянутое в (24) вхождение x), то из (1) следует, что $y^{s'} \subset k(\tilde{e})$.

Из равенства $e' = e^\theta$ следует, что $\exists\, z \in Var(e)$: вышеупомянутое вхождение z' в e' входит в подтерм $z^\theta \subseteq e^\theta = e'$. Следовательно, $(z')^s \subseteq (z^\theta)^s = z^{s'} \subseteq e^{s'}$.

Таким образом, получаем:

$$y^{s'} \subset k(\tilde{e}) \subseteq (z')^s \subseteq z^{s'} \subseteq e^{s'}. \tag{25}$$

Как и в предыдущем пункте, на основании (25) заключаем, что терм e содержит вхождения переменных y и z, обладающие следующим свойством: $y^{s'} \subset z^{s'}$, откуда для данных вхождений следует включение $y \subset z$, что невозможно.

2) Если второе утверждение в (19) неверно, то из второго утверждения в (18) следует, что

$$\left.\begin{array}{l} \alpha \text{ имеет вид } c!e, \text{ где } e \in \langle P \rangle_s, \\ \exists \text{ вхождение } x \text{ в } e^s, \text{ которое не содержится} \\ \text{ни в каком подтерме вида } k(\ldots) \subseteq e^s, \text{ где } k \in E_{\mathbf{K}}. \end{array}\right\} \tag{26}$$

Поскольку $e \in \langle P \rangle_s$, то упомянутое в (26) вхождение x в e^s содержится в подтерме вида y^s терма e^s, где y – некоторая переменная из $[P]_s$. Согласно первому утверждению в (18), это вхождение x в y^s содержится в подтерме вида $k(\ldots) \subseteq y^s \subseteq e^s$, где $k \in E_{\mathbf{K}}$. Но это противоречит (26). ■

Теорема 4.

Пусть заданы РП $\mathcal{P}$ такой, что $Var(\mathcal{P})_{\mathbf{C}} = \{\circ\}$, ПП $P \in \mathcal{P}$, состояния $s, s' \in \Sigma_{\mathcal{P}}$, такие, что $s \xrightarrow{\alpha_P} s'$, и подмножество $E \subseteq \langle \mathcal{P} \rangle_0$.

Тогда $\forall\, k \in E_{\mathbf{K}} : k \neq public\_key(\ldots)$ верна импликация

$$s \models \varphi \Rightarrow s' \models \varphi, \quad \text{где } \varphi = \{E \perp_{\mathbf{K}} P, k^{-1}[P] \subseteq k^{-1}[\circ]\}.$$

Доказательство.

По теореме 3, из $s \models E \perp_{\mathbf{K}} P$ следует $s' \models E \perp_{\mathbf{K}} P$. Таким образом, для доказательства теоремы 4 достаточно доказать импликацию

$$s \models \varphi \;\; \Rightarrow \;\; s' \models k^{-1}[P] \subseteq k^{-1}[\circ]. \tag{27}$$

Если (27) неверно, то α – не посылка, $[P]_{s'}$ имеет вид $[P]_s \cup Var(e')$, и для некоторого терма $e \in k^{-1}[P]^{s'}$ выполняется свойство

$$e \notin k^{-1}[\circ]_s, \text{ т.е. } \exists\, x \in [P]_{s'} : k(e) \subseteq x^{s'}, \forall\, \dot{e} \in [\circ]_s \;\; k(e) \not\subseteq \dot{e}. \tag{28}$$

Из предположения $s \models \varphi$ и из (28) следует, что $x \in Var(e')$, откуда получаем: $k(e) \subseteq x^{s'} \subseteq (e')^{s'}$.

Рассмотрим по отдельности каждый из двух возможных видов α.

1) $\alpha = ?e'$, в этом случае $k(e) \subseteq (e')^{s'} \in [\circ]_s$. Полагая в (28) терм $\dot{e}$ равным $(e')^{s'}$, получаем противоречие.

2) $\alpha = (e' := e'')$, в этом случае $e'' \in \langle P \rangle_s$, $(e')^{s'} = (e'')^s$, поэтому

$$k(e) \subseteq (e'')^s. \tag{29}$$

Докажем, что $k \notin e''$ и $k \in E_{\mathbf{X}}$.

Предположим, что $k \in e''$. По условию теоремы, $k \neq public\_key(\ldots)$. Если $k = shared\_key(\ldots)$, то, по определению выполнения ПП в пункте 2.2.5, должно быть верно соотношение $Agent(P) \in k \in E_{\mathbf{K}}$, которое противоречит первому условию свойства $s \models E \perp_{\mathbf{K}} P$. Напомним, что данное свойство имеет вид: $\forall\, \tilde{e} \in E \;\; Agent(P) \notin \tilde{e}$, и

$$\left.\begin{array}{l} \forall\, x \in E_{\mathbf{X}}, \forall\, y \in [P]_s \;\; x \perp_{\mathbf{K},E} y^s \\ \forall\, x \in E_{\mathbf{X}}, \forall\, \tilde{e} \in [\circ]_s \;\; x \perp_{\mathbf{K},E} \tilde{e} \end{array}\right\} \tag{30}$$

Следовательно, $k \in E_{\mathbf{X}}$, поэтому из $k \in e'' \in \langle P \rangle_s$ следует $k \in [P]_s$. Однако, полагая в первом соотношении в (30) x и y равными k, получаем утверждение $k \perp_{\mathbf{K},E} k$, которое является ложным, согласно определению (6).

Аналогично доказательству импликации (21) $\Rightarrow$ (22) в теореме 3, можно доказать, что из свойств $k \notin e''$ и (29) следует, что

$$\exists\, y \in Var(e'') \subseteq [P]_s : k(e) \subseteq y^s. \tag{31}$$

По предположению, $s \models k^{-1}[P] \subseteq k^{-1}[\circ]$, т.е. $k^{-1}[P]^s \subseteq k^{-1}[\circ]_s$. Из (31) следует, что $e \in k^{-1}[P]^s$. Следовательно, $e \in k^{-1}[\circ]_s$, что противоречит предположению (28). ■

Следующая теорема является усилением теоремы 4. В ней утверждается, что в условиях теоремы 4 нижняя и верхняя границы на множество ШС, содержащихся в открытом канале, и зашифрованных на защищённых ключах, не изменяются при выполнении действий ПП P.

Теорема 5.

Пусть заданы РП $\mathcal{P}$ такой, что $Var(\mathcal{P})_{\mathbf{C}} = \{\circ\}$, ПП $P \in \mathcal{P}$, состояния $s, s' \in \Sigma_{\mathcal{P}}$, такие, что $s \xrightarrow{\alpha_P} s'$, и подмножества $E \subseteq \langle\mathcal{P}\rangle_0$, $E', E'' \subseteq Tm$.

Тогда $\forall\, k \in E_{\mathbf{K}} : k \neq public\_key(\ldots)$ верна импликация

$$s \models \varphi \Rightarrow s' \models \varphi, \text{ где } \varphi = \left\{ \begin{array}{l} E \perp_{\mathbf{K}} P, k^{-1}[P] \subseteq k^{-1}[\circ] \\ E' \subseteq k^{-1}[\circ] \subseteq E'' \end{array} \right\}. \qquad (32)$$

Доказательство.

По теореме 4, для доказательства (32) достаточно доказать, что

$$s \models \varphi \;\; \Rightarrow \;\; s' \models \{E' \subseteq k^{-1}[\circ], k^{-1}[\circ] \subseteq E''\}. \qquad (33)$$

- Утверждение $s' \models E' \subseteq k^{-1}[\circ]$ следует из того, что $[\circ]_s \subseteq [\circ]_{s'}$.
- Докажем утверждение $s' \models k^{-1}[\circ] \subseteq E''$. Если бы оно было неверным, то было бы верно неравенство $[\circ]_s \neq [\circ]_{s'}$. Это возможно только если α имеет вид $!e$, где $e \in \langle P\rangle_s$, и

  $$[\circ]_{s'} = [\circ]_s \cup \{e^s\}, \text{ причем } \exists\, e' \notin (E'')^s \supseteq k^{-1}[\circ]_s : k(e') \subseteq e^s. \qquad (34)$$

  Так же, как и в теореме 4, доказываем, что $k \in E_{\mathbf{X}}$, и что из $e \in \langle P\rangle_s$ следует соотношение $k \notin e$.

  Аналогично доказательству импликации (21) $\Rightarrow$ (22) в теореме 3, можно доказать, что из $k \notin e$ и $k(e') \subseteq e^s$ следует, что

  $$\exists\, x \in Var(e) \subseteq [P]_s :\; k(e') \subseteq x^s,$$

  поэтому $e' \in k^{-1}[P]^s$. Отсюда, используя предположение $s \models \varphi$, следствием которого является включение $k^{-1}[P]^s \subseteq k^{-1}[\circ]_s$, получаем: $e' \in k^{-1}[\circ]_s$, что противоречит (34). ■

2.5. Теорема для доказательства свойства соответствия

Теорема, излагаемая в этом параграфе, может использоваться для доказательства **свойства соответствия** протоколов аутентификации, которое имеет следующий смысл: если один из участников протокола аутентификации после выполнения этого протокола пришел к выводу, что другой участник этого протокола является подлинным (т.е. объявленные им свое имя и параметры совпадают с его реальными именем и параметрами), то это действительно так. Доказываемая ниже теорема применяется для обоснования того, что если

- РП $\mathcal{P}$ использует для взаимодействия только открытый канал $\circ$, и
- в некотором состоянии $s \in \Sigma_{\mathcal{P}}$ в этом канале содержится сообщение, содержащее подтерм вида $k(e)$, где ключ k недоступен в этом состоянии для некоторого ПП P, входящего в $\mathcal{P}$,

то в некотором состоянии $s' <_\pi s$ другой ПП $P' \neq P$ из $\mathcal{P}$ послал в открытый канал $\circ$ сообщение, содержащее тот же самый подтерм $k(e)$.

В параграфах 3.2 и 3.3 мы рассматриваем примеры применения данной теоремы для верификации КП Yahalom и КП передачи ШС между несколькими агентами.

Теорема 6.

Пусть заданы РП $\mathcal{P}$, такой, что $Var(\mathcal{P})_{\mathbf{C}} = \{\circ\}$, ПП $P \in \mathcal{P}$, множество $E \subseteq \langle\mathcal{P}\rangle_0$, не содержащее открытых ключей, и состояние $s \in \Sigma_{\mathcal{P}}$, причем

- $s \models E \perp_{\mathbf{K}} P$, и
- $[\circ]_s$ содержит терм с подтермом $k(e)$, где $k \in E_{\mathbf{K}}$.

Тогда для каждого пути π из начального состояния 0 РП $\mathcal{P}$ в состояние s существует ПП $P' \in \mathcal{P} \setminus \{P\}$, такой, что π содержит ребро вида

$$\dot{s} \xrightarrow{(!\dot{e})_{P'}} s', \quad \text{где } k(e) \subseteq \dot{e}^{\dot{s}}. \tag{35}$$

Доказательство.

Обозначим записью s' первое состояние на пути π, такое, что $[\circ]_{s'}$ содержит терм e' с подтермом $k(e)$. Т.к. $[\circ]_0 = \emptyset$, то $s' \neq 0$.

Пусть ребро на пути π с концом в s' имеет вид $\dot{s} \overset{\alpha_{P'}}{\rightarrow} s'$. Т.к. $e' \notin [\circ]_{\dot{s}}$, то $\alpha = !\dot{e}$, где $\dot{e}^{\dot{s}} = e'$. Если $P' \neq P$, то теорема доказана.

Докажем, что другой возможный случай ($P' = P$) невозможен.

Предположим, что $P' = P$, т.е. $\dot{s} \xrightarrow{(!\dot{e})_P} s'$.

Докажем, что $k \in E_{\mathbf{X}}$.

Если это неверно, т.е. k – разделяемый ключ, то, согласно определению выполнения ПП в пункте 2.2.5, должно быть $Agent(P) \in k \in E_{\mathbf{K}}$, что противоречит предположению $\forall\, \tilde{e} \in E \;\; Agent(P) \notin \tilde{e}$.

Из $s \models E \perp_{\mathbf{K}} P$ следует, что $\dot{s} \models E \perp_{\mathbf{K}} P$, откуда получаем $k \notin [P]_{\dot{s}}$, т.к. если $k \in [P]_{\dot{s}}$, то, согласно (5), вхождение k в $k^{\dot{s}} = k$ должно содержаться в подтерме вида $k'(\ldots)$, что невозможно.

Из $k \notin [P]_{\dot{s}}$ и из условия $\dot{e} \in \langle P\rangle_{\dot{s}}$, которое верно согласно (7)(a), следует, что $k \notin \dot{e}$.

Аналогично доказательству импликации (21) $\Rightarrow$ (22) в теореме 3, можно доказать, что из свойств $k(e) \subseteq e' = \dot{e}^{\dot{s}}$ и $k \notin \dot{e}$ следует, что

$$\exists\, x \in Var(\dot{e}) \subseteq [P]_{\dot{s}} : k(e) \subseteq x^{\dot{s}} \in [P]^{\dot{s}}. \tag{36}$$

Обозначим записью s'' первое состояние на пути π, такое, что $[P]^{s''}$ содержит терм с подтермом $k(e)$, т.е.

$$\exists\, x \in [P]_{s''} : k(e) \subseteq x^{s''}. \tag{37}$$

Из (36) следует, что s'' находится на пути π левее s'. Нетрудно видеть, что $s'' \neq 0$, поэтому на пути π существует ребро вида $\ddot{s} \xrightarrow{\alpha_{P''}} s''$. Из выбора s'' следует, что $x \notin [P]_{\ddot{s}}$, поэтому $P'' = P$, и возможны два случая:

1) $\alpha = ?\ddot{e}, x \in Var(\ddot{e}), \ddot{e}^{s''} \in [\circ]_{\ddot{s}}$,

   т.к. $k(e) \subseteq x^{s''} \subseteq \ddot{e}^{s''} \in [\circ]_{\ddot{s}}$, то получаем противоречие с выбором s' как самого первого состояния на пути π, такого, что $[\circ]_{s'}$ содержит терм e' с подтермом $k(e)$: состояние $\ddot{s}$ имеет аналогичное свойство, и находится левее s',

2) $\alpha = (\ddot{e} := \tilde{e}), x \in Var(\ddot{e}), \tilde{e} \in \langle P \rangle_{\ddot{s}}, \ddot{e}^{s''} = \tilde{e}^{\ddot{s}}$,

   поскольку

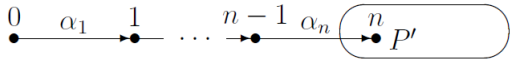
   - $k(e) \subseteq x^{s''} \subseteq \ddot{e}^{s''} = \tilde{e}^{\ddot{s}}$ и
   - $\tilde{e}$ не содержит k, т.к. выше было установлено, что $k \notin [P]_{\dot{s}}$, поэтому, учитывая свойство $\ddot{s} \leq \dot{s}$, из которого следует включение $[P]_{\ddot{s}} \subseteq [P]_{\dot{s}}$, получаем: $k \notin [P]_{\ddot{s}}$, и, следовательно, терм $\tilde{e} \in \langle P \rangle_{\ddot{s}}$ тоже не содержит k,

   то, аналогично доказательству импликации (21) $\Rightarrow$ (22) в теореме 3, можно доказать, что $\exists\, y \in [P]_{\ddot{s}} : k(e) \subseteq y^{\ddot{s}}$, что противоречит выбору s'' как самого первого состояния на пути π со свойством (37): $\ddot{s}$ имеет аналогичное свойство, и находится левее s''. ■

2.6. Схемы распределенных процессов

2.6.1. Префиксные последовательные процессы

Будем говорить, что ПП P является **префиксным**, если он имеет вид

$P =$ 

(38)

т.е. в P имеется совокупность вершин, занумерованных натуральными числами $0, 1, \ldots, n$ $(n \geq 1)$, причем $Init(P) = 0$, $\forall\, i = 0, \ldots, n-1$ из

вершины i выходит ровно одно ребро с концом $i+1$ и меткой α_i. Запись P' в (38) обозначает подграф графа P, состоящий из вершин и ребер графа P, за исключением вершин $0, \ldots, n-1$ и связанных с ними ребер.

Подграфы $0 \overset{\alpha_1}{\to} 1 \overset{\alpha_2}{\to} \ldots \overset{\alpha_n}{\to} n$ и P' графа (38) будем называть **префиксом** и **постфиксом** ПП P соответственно, и обозначаются записями $\mathit{Pref}(P)$ и $\mathit{Post}(P)$ соответственно. Последняя вершина в $\mathit{Pref}(P)$ называется **конечной** вершиной этого префикса. Если $\mathit{Post}(P)$ состоит из одной вершины, то он обозначается символом $\mathbf{0}$.

Если ПП P имеет вид (38), то будем обозначать этот факт записью

$$P = \alpha_1; \ldots; \alpha_n; P'. \tag{39}$$

2.6.2. Понятие схемы распределенного процесса

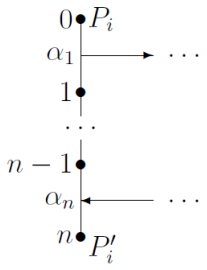
Пусть задан РП $\mathcal{P}$, и $\forall\, P \in \mathcal{P}$ ПП P – префиксный, причем для каждой посылки (каждого приема) в $\mathit{Pref}(P)$

- предполагаемым получателем (отправителем) того сообщения, которое посылается (принимается) при выполнении этого действия, является определенный ПП P' из $\mathcal{P}$, и
- действие ПП P', соответствующее приему (посылке) этого сообщения, находится в $\mathit{Pref}(P')$.

Эти зависимости между действиями можно выразить в виде **схемы** РП $\mathcal{P}$, которая имеет следующий вид:

- каждый ПП $P \in \mathcal{P}$ представляется в этой схеме **нитью**, т.е. вертикальной линией, на которой выделены точки, соответствующие вершинам из $\mathit{Pref}(P)$, верхняя точка соответствует $Init(P)$, причем
  - у каждой точки указан номер соответствующей вершины,
  - рядом с верхней точкой указывается имя ПП P,
  - если $\mathit{Post}(P) = P' \neq \mathbf{0}$, то у нижней точки указано P',
  - рядом с каждым отрезком l, соединяющим соседние точки нити, указана метка α_l ребра из $\mathit{Pref}(P_i)$, соответствующего l,
- для каждого отрезка l, соединяющего соседние точки нити, если α_l – посылка сообщения, то в схеме присутствует стрелка, начало которой лежит на l, а конец – на отрезке l', таком, что $\alpha_{l'}$ – действие соответствующего ПП $P_{i'}$ по приему этого сообщения.

Например если $P_i = \alpha_1;\ \ldots\ \alpha_n; P_i'$, где α_1 – посылка, а α_n – прием, то процессу P_i соответствует нить



(40)

Отметим, что стрелки от посылок к приемам сообщений изображают лишь желаемую связь между этими действиями, но отнюдь не реальную связь: возможно что посланное сообщение будет принято из канала совсем не тем процессом, для которого оно было предназначено.

В целях большей наглядности будем использовать следующее соглашение в обозначениях переменных.

- Будем указывать горизонтальную черту над любым обозначением какой-либо переменной x, если она рассматривается как элемент множества вида $\bar{X}(P)$ (т.е. эта переменная обозначается $\bar{x}$).
- Если P – ПП вида (39), и переменная $x \in \hat{X}(P)$ входит в метку α_i i–го отрезка нити ПП P, причем i – первая метка в (39), в которую входит x (т.е. $\forall\, i' = 1, \ldots, i-1$ x не входит в $\alpha_{i'}$), то над этим вхождением x указывается уголок (т.е. это вхождение обозначается записью $\hat{x}$). Данные обозначения переменных используются также в записях вида (39).

2.6.3. Примеры схем распределенных процессов

1) Первый пример – это РП $\mathcal{P}_1 = \{A, B\}$, являющийся моделью передачи от A к B сообщения x по каналу c_{AB}, где только A и B знают имя этого канала, т.е. $c_{AB} = shared\_channel(A, B)$.

   Данный РП работает следующим образом:

   - A посылает B сообщение x по каналу c_{AB},
   - B принимает из канала c_{AB} сообщение и записывает его в переменную y, после чего ведет себя так же, как процесс P.

   ПП A и B имеют следующий вид: $A = c_{AB}!x; \mathbf{0}$, $B = c_{AB}?\hat{y}; P$.

Схема РП $\mathcal{P}_1$ имеет следующий вид:

(41)

2) Второй пример – это РП $\mathcal{P}_2 = \{A, B\}$, являющийся моделью передачи от A к B ШС $k_{AB}(x)$ по открытому каналу $\circ$. Предполагается, что A и B имеют общий секретный ключ k_{AB}, на котором они могут шифровать и дешифровать сообщения, используя симметричную систему шифрования, причем только A и B знают k_{AB}, т.е. $k_{AB} = shared\_key(A, B)$.

   Данный РП работает следующим образом:

   - A посылает B ШС $k_{AB}(x)$ по каналу $\circ$,
   - B принимает из канала $\circ$ сообщение $k_{AB}(x)$, дешифрует его, записывает извлеченное сообщение x в переменную y, после чего ведет себя так же, как процесс P.

   ПП A и B имеют следующий вид: $A = !k_{AB}(x); \mathbf{0}$, $B = ?k_{AB}(\hat{y}); P$.

   Схема РП $\mathcal{P}_2$ имеет следующий вид:

(42)

3) Третий пример – это РП $\mathcal{P}_3 = \{A, B, J\}$, являющийся моделью передачи от A к B одного сообщения x по скрытому каналу $\bar{c}$ при помощи **доверенного посредника** J, где A и J (B и J) взаимодействуют по скрытому каналу c_{AJ} (c_{BJ}), причем только A и J (B и J) знают имя c_{AJ} (c_{BJ}), т.е. $c_{AJ} = shared\_channel(A, J)$, $c_{BJ} = shared\_channel(B, J)$.

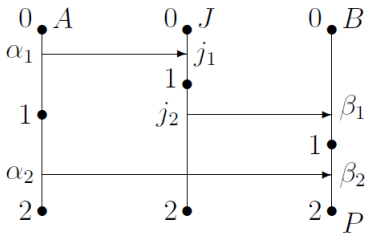
   Данный РП работает следующим образом:

   - A посылает J имя скрытого канала $\bar{c}$ (которое сначала знает только он один) по каналу c_{AJ},
   - J посылает B полученное имя канала $\bar{c}$ по каналу c_{BJ},
   - A посылает B сообщение x по каналу $\bar{c}$,
   - B принимает из канала $\bar{c}$ сообщение и записывает его в переменную y, после чего ведет себя так же, как процесс P.

ПП A, B и J определяются следующим образом:

$$\begin{array}{llll} A = \alpha_1; \alpha_2; \mathbf{0}, & \text{где} & \alpha_1 = c_{AJ}!\bar{c}, & \alpha_2 = \bar{c}!x, \\ J = j_1; j_2; \mathbf{0}, & \text{где} & j_1 = c_{AJ}?\hat{u}, & j_2 = c_{BJ}!u, \\ B = \beta_1; \beta_2; P, & \text{где} & \beta_1 = c_{BJ}?\hat{v}, & \beta_2 = v?\hat{y}. \end{array} \tag{43}$$

Схема РП $\mathcal{P}_3$ имеет следующий вид:



(44)

4) Четвертый пример – это РП $\mathcal{P}_4 = \{A, B, J\}$ (называемый **протоколом Wide-Mouth Frog (WMF)**), являющийся моделью передачи от A к B ШС $\bar{k}(x)$ по открытому каналу $\circ$ при помощи доверенного посредника J, с которым A и B взаимодействуют по открытому каналу. A создает секретный ключ $\bar{k}$, посылает B этот ключ в зашифрованном виде черед доверенного посредника J, и затем посылает B ШС $\bar{k}(x)$.

Предполагается, что A и J (B и J) имеют общий секретный ключ k_{AJ} (k_{BJ}), на котором они могут шифровать и дешифровать сообщения, используя симметричную систему шифрования, причем только A и J (B и J) знают k_{AJ} (k_{BJ}), т.е. $k_{AJ} = shared\_key(A, J)$, $k_{BJ} = shared\_key(B, J)$.

Данный РП работает следующим образом.

- A создает секретный ключ $\bar{k}$ (сначала только A знает этот ключ) и посылает доверенному посреднику J ШС $k_{AJ}(\bar{k})$ по каналу $\circ$, после чего A посылает B ШС $\bar{k}(x)$ по каналу $\circ$,
- J принимает сообщение от A, дешифрует его, затем шифрует извлеченный ключ $\bar{k}$ на ключе k_{BJ}, и посылает B ШС $k_{BJ}(\bar{k})$ по каналу $\circ$,
- B извлекает из полученного сообщения от J ключ $\bar{k}$, и затем использует этот ключ для извлечения из полученного сообщения от A сообщения x, записывает его в переменную y, после чего ведет себя так же, как процесс P.

ПП A, B и J определяются следующим образом:

$$\begin{array}{llll} A = \alpha_1; \alpha_2; \mathbf{0}, & \text{где} & \alpha_1 = !k_{AJ}(\bar{k}), & \alpha_2 = !\bar{k}(x), \\ J = j_1; j_2; \mathbf{0}, & \text{где} & j_1 = ?k_{AJ}(\hat{u}), & j_2 = !k_{BJ}(u), \\ B = \beta_1; \beta_2; P, & \text{где} & \beta_1 = ?k_{BJ}(\hat{v}), & \beta_2 = ?v(\hat{y}). \end{array} \tag{45}$$

Схема РП $\mathcal{P}_4$ имеет вид (44).

2.7. Графы переходов распределенных процессов

В этом параграфе рассматриваются РП, состоящие из конечного числа ПП. Для наглядного представления выполнения таких РП вводится понятие **графа переходов** РП. Выполнение РП м.б. представлено как обход вершин ГП, соответствующего этому РП.

Ниже в этом параграфе символ $\mathcal{P}$ обозначает РП, состоящий из конечного числа ПП, каждый из которых отличен от $P_\dagger$.

2.7.1. Понятие графа переходов распределенного процесса

Графом переходов (ГП) РП $\mathcal{P} = \{P_1, \ldots, P_n\}$ называется граф $G_\mathcal{P}$,

- каждая вершина которого представляет собой список

$$at = (at_1, \ldots, at_n), \quad \text{где } \forall\, i = 1, \ldots, n \;\; at_i \in P_i,$$

- каждое ребро которого имеет вид

$$(at_1, \ldots, at_n) \xrightarrow{\alpha_{P_i}} (at'_1, \ldots, at'_n) \quad (i \in \{1, \ldots, n\}), \tag{46}$$

где P_i содержит ребро $at_i \overset{\alpha}{\to} at'_i$, и $at_{i'} = at'_{i'}$ при $i' \neq i$.

Вершина $(Init(P_1), \ldots, Init(P_n))$ ГП $G_\mathcal{P}$ называется **начальной** вершиной ГП $G_\mathcal{P}$, и обозначается записью $Init(G_\mathcal{P})$.

Нетрудно доказать, что если графы всех ПП, входящих в $\mathcal{P}$, ациклични, то $G_\mathcal{P}$ тоже ацикличен.

Для каждого состояния $s \in \Sigma_\mathcal{P}$ компонента at_s состояния s может рассматриваться как вершина ГП $G_\mathcal{P}$.

Пусть задано некоторое выполнение РП $\mathcal{P}$. Если при этом выполнении порождается последовательность состояний $s_0, s_1, \ldots, s_n$, то из определения отношения перехода (9) следует, что этой последовательности соответствует путь в ГП $G_\mathcal{P}$, который можно рассматривать как наглядное представление выполнения РП $\mathcal{P}$:

$$Init(G_\mathcal{P}) = at_{s_0} \xrightarrow{(\alpha_1)_{P_{i_1}}} at_{s_1} \xrightarrow{(\alpha_2)_{P_{i_2}}} \ldots \xrightarrow{(\alpha_n)_{P_{i_n}}} at_{s_n}$$

Напомним, что $\mathcal{P}_\dagger = \{\mathcal{P}, P_\dagger\}$. ГП $G_{\mathcal{P}_\dagger}$ можно рассматривать как граф, получаемый добавлением к $G_\mathcal{P}$ рёбер $at \xrightarrow{\alpha_{P_\dagger}} at$, где $at \in G_\mathcal{P}$, $\alpha \in Act$.

Вершина $at \in G_\mathcal{P}$ называется **достижимой**, если $\exists\, s \in \Sigma_{\mathcal{P}_\dagger}\colon at = at_s$.

Ребро ГП $G_\mathcal{P}$ называется **реализуемым**, если оно лежит на пути, соответствующем какому-либо выполнению РП $\mathcal{P}_\dagger$.

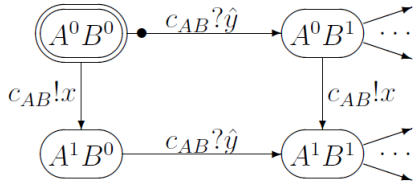
В изображении ГП $G_\mathcal{P}$ будут использоваться следующие соглашения:

- каждая вершина $at = (at_1, \ldots, at_n)$ ГП $G_\mathcal{P}$ обозначается овалом, внутри которого указан список $at_1 \ldots at_n$ компонентов at,
- начальная вершина $Init(G_\mathcal{P})$ изображается двойным овалом,
- черный кружочек на каком–либо ребре ГП $G_\mathcal{P}$ означает нереализуемость этого ребра (обоснование нереализуемости рёбер должно проводиться специальными рассуждениями),
- в целях сокращения обозначений, метка ребра $at \xrightarrow{\alpha_P} at'$ ГП $G_\mathcal{P}$ может обозначаться просто действием α в этой метке (без указания ПП P, выполняющего действие α при данном переходе).

2.7.2. Примеры графов переходов распределенных процессов

В этом пункте приводятся примеры ГП для РП, представленных схемами из пункта 2.6.3. Будем использовать следующее соглашение: если A – имя ПП, входящего в какой-либо из этих РП, и i – номер точки на нити, соответствующей этому ПП, то вершина графа A, соответствующая этой точке, обозначается записью A^i.

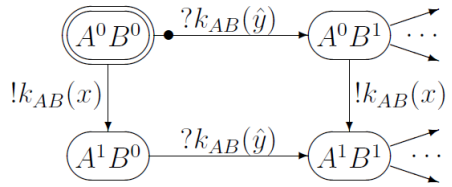
1) ГП для РП $\mathcal{P}_1$, описываемого схемой (41):



(47)

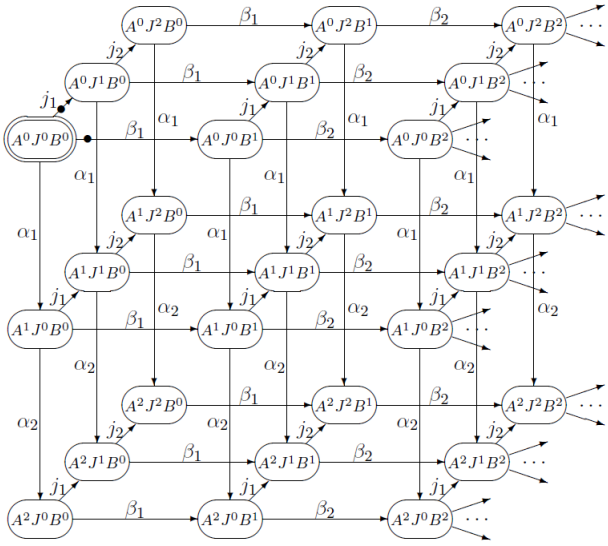
где наклонные стрелки обозначают ребра ГП, выходящие из соответствующих вершин, а также части ГП, достижимые после прохода по этим ребрам, которые не представлены в данном изображении ГП, это соглашение будет использоваться также и в нижеследующих примерах ГП.

2) ГП для РП $\mathcal{P}_2$, описываемого схемой (42):



(48)

3) ГП для РП $\mathcal{P}_3$ и $\mathcal{P}_4$, описываемых схемой (44):



(49)

2.7.3. Истинность формул в вершинах графов переходов распределенных процессов

Пусть задан РП $\mathcal{P}$. Для каждой вершины $at \in G_{\mathcal{P}}$ и каждой формулы $\varphi \in Fm$ запись $at \models \varphi$ обозначает утверждение

$$\varphi \text{ истинна в вершине } at$$

которое считается верным, если $\forall\, s \in \Sigma_{\mathcal{P}_\dagger}\ \ at_s = at \Rightarrow s \models \varphi$.

Нетрудно доказать, что ГП $G_{\mathcal{P}_i}$, где $\mathcal{P}_i$ $(i = 1, \ldots, 4)$ – РП, определенные в пункте 2.6.3, обладают следующими свойствами:

$$\left.\begin{array}{l} Init(G_{\mathcal{P}_1}) \models \{\varphi_1, [c_{AB}] = \emptyset\} \\ Init(G_{\mathcal{P}_2}) \models \{\varphi_2, k_{AB}^{-1}[\circ] = \emptyset\} \\ Init(G_{\mathcal{P}_3}) \models \{\varphi_3, [c_{AJ}] = \emptyset, [c_{BJ}] = \emptyset, [\bar{c}] = \emptyset\} \\ Init(G_{\mathcal{P}_4}) \models \{\varphi_4, k_{AJ}^{-1}[\circ] = \emptyset, k_{BJ}^{-1}[\circ] = \emptyset, \bar{k}^{-1}[\circ] = \emptyset\} \end{array}\right\} \tag{50}$$

где

$$\varphi_1 = \{c_{AB}\} \perp_{\mathbf{C}} P_\dagger$$
$$\varphi_2 = \{\{k_{AB}\} \perp_{\mathbf{K}} P_\dagger, k_{AB}^{-1}[P_\dagger] \subseteq k_{AB}^{-1}[\circ]\}$$
$$\varphi_3 = \{c_{AJ}, c_{BJ}, \bar{c}\} \perp_{\mathbf{C}} P_\dagger$$
$$\varphi_4 = \left\{ \begin{array}{l} \{k_{AJ}, k_{BJ}, \bar{k}\} \perp_{\mathbf{K}} P_\dagger \\ k_{AJ}^{-1}[P_\dagger] \subseteq k_{AJ}^{-1}[\circ], k_{BJ}^{-1}[P_\dagger] \subseteq k_{BJ}^{-1}[\circ], \bar{k}^{-1}[P_\dagger] \subseteq \bar{k}^{-1}[\circ] \end{array} \right\}$$

Действительно, для любого РП $\mathcal{P}$ состояние $s \in \Sigma_{\mathcal{P}_\dagger}$ обладает свойством $at_s = Init(G_{\mathcal{P}})$, если $s = 0$, или существует путь из 0 в s с метками ребер вида $\alpha_{P_\dagger}$, и для каждой из формул $\psi_i = \{\varphi_i, \ldots\}$ в (50)

- истинность ψ_i в начальном состоянии РП $(\mathcal{P}_i)_\dagger$ следует из определений понятия начального состояния и ПП $P_\dagger$, и
- истинность ψ_i в состоянии s, в которое существует путь из 0 с метками ребер вида $\alpha_{P_\dagger}$, обосновывается утверждением

$$\forall\, s', s'' \in \Sigma_{\mathcal{P}_\dagger} : s' \xrightarrow{\alpha_{P_\dagger}} s'' \quad (s' \models \psi_i \;\; \Rightarrow \;\; s'' \models \psi_i) \tag{51}$$

  которое следует из теоремы 2 ($i = 1, 3$), или теоремы 5 ($i = 2, 4$).

Мы будем использовать соотношения (50) в излагаемом ниже решении задачи верификации некоторых свойств РП $\mathcal{P}_i$ $(i = 1, \ldots, 4)$.

3. Методы верификации криптографических протоколов

Излагаемые в этом параграфе методы верификации КП основаны на представлении КП в виде РП. Для доказательства свойств РП используются теоремы из предыдущего параграфа. Первый из излагаемых ниже методов использует понятие ГП, а второй основан на теореме 6 и наиболее подходит для верификации КП аутентификации.

В этом параграфе предполагается что символ $\mathcal{P}$ обозначает РП, не содержащий ПП противника $P_\dagger$.

3.1. Верификация на основе графов переходов

3.1.1. Описание метода

Некоторые свойства РП могут выражаться формулами, связаными с достижимыми вершинами соответствующих ГП. Например, одно из

свойств РП $\mathcal{P}_i$ $(i = 1, \ldots, 4)$, определенных в пункте 2.6.3, имеет такой вид:

$$\begin{array}{l}\text{если достижимая вершина } at = (at_A, at_B) \text{ или } (at_A, at_J, at_B) \\ \text{ГП } G_{\mathcal{P}_i} \text{ такова, что } at_B = 1 \ (i = 1, 2) \text{ или } at_B = 2 \ (i = 3, 4), \\ \text{то } at \models x = y.\end{array} \qquad (52)$$

Данное свойство называется **целостностью** и имеет следующий смысл:

- если выполнение РП $(\mathcal{P}_i)_\dagger = \{A, B, P_\dagger\}$ или $\{A, J, B, P_\dagger\}$ достигло состояния, где принимающий ПП B заканчивает часть своего выполнения, связанную с приемом сообщения от передающего ПП A,
- то при любом противодействии противника $P_\dagger$ передаваемое сообщение x в передающем ПП A совпадает с тем значением, которое будет иметь переменная y в принимающем ПП B.

Если свойство РП $\mathcal{P}$ имеет вид $at \models \varphi$, где at – некоторая достижимая вершина ГП $G_{\mathcal{P}}$, то один из методов верификации этого свойства заключается в следующем:

- для каждой достижимой вершины at', находящейся на каком-либо пути из $Init(G_{\mathcal{P}})$ в at, вычисляется формула $\varphi_{at'}$, истинная в at', и
- проверяется свойство $\varphi_{at} \leq \varphi$.

Необходимость вычисления вышеупомянутых формул для всех вершин на путях из $Init(G_{\mathcal{P}})$ в at связана с тем, что для вычисления φ_{at} необходимо знать формулы $\varphi_{at'}$ для каждой достижимой вершины at', из которой существует реализуемое ребро в at, и т.д.

Метод вычисления формулы φ_{at} имеет следующий вид:

- если вычислена формула $\varphi_{at'}$ для какой-либо достижимой вершины at', такой, что существует ребро ГП $G_{\mathcal{P}}$ вида $at' \overset{\alpha}{\to} at$, то вычисляется формула $\alpha(\varphi_{at'})$, смысл которой заключается в следующем: если в текущем состоянии s была верна формула $\varphi_{at'}$, и $s \overset{\alpha}{\to} s'$, то формула $\alpha(\varphi_{at'})$ верна в состоянии s', а также в каждом состоянии, в которое существует путь из s' с метками ребер вида $\alpha_{P_\dagger}$,
- искомая формула φ_{at} определяется как аналог дизъюнкции всех формул вида $\alpha(\varphi_{at'})$.

Для начальной вершины $at^0 = Init(G_{\mathcal{P}})$ соответствующая формула φ_{at^0} предполагается заданной. Например, для каждого из РП $\mathcal{P}_i$ ($i = 1, \ldots, 4$), определенных в пункте 2.6.3, в качестве такой формулы можно взять формулу из соответствующего соотношения в (50).

Ниже излагаются примеры применения данного метода для решения задач верификации РП, в которых используются защищенные каналы ($\mathcal{P}_1$ и $\mathcal{P}_3$), или защищенные ключи ($\mathcal{P}_2$ и $\mathcal{P}_4$). Перед решением задач верификации данных РП мы изложим теоремы, которые будут использоваться для решения этих задач.

3.1.2. Теоремы, используемые для верификации распределенных процессов с защищенными каналами

В этом пункте будем использовать следующее обозначение:

$$\begin{array}{l} \forall\, E \subseteq Tm, \forall\, \alpha \in Act, \forall\, c \in Channels \\ E_{c,\alpha} = E \cup \{e\},\ \text{если}\ \alpha = c!e,\ \text{и}\ E_{c,\alpha} = E,\ \text{иначе}. \end{array} \tag{53}$$

Теорема 7.

Пусть заданы РП $\mathcal{P}$, подмножество $E \subseteq \langle \mathcal{P} \rangle_0$, состояния $s, s' \in \Sigma_{\mathcal{P}_\dagger}$ такие, что $s \xrightarrow{\alpha_P} s'$, где $P \in \mathcal{P}$, и если $\alpha = c!e$, то верна импликация

$$c^s \notin E \ \ \Rightarrow \ \ Var(e^s) \cap E = \emptyset. \tag{54}$$

Тогда $\forall\, E', E'' \subseteq Tm$, $\forall\, c \in E_{\mathbf{C}}$ верна импликация

$$s \models \left\{ \begin{array}{l} E \perp_{\mathbf{C}} P_\dagger \\ E' \subseteq [c] \subseteq E'' \end{array} \right\} \Rightarrow s' \models \left\{ \begin{array}{l} E \perp_{\mathbf{C}} P_\dagger \\ E'_{c,\alpha} \subseteq [c] \subseteq E''_{c,\alpha} \end{array} \right\}$$

Доказательство.

Согласно (4), значение формулы $E \perp_{\mathbf{C}} P_\dagger$ в s зависит только от множеств $[P_\dagger]^s$ и $[c]^s$ ($\forall\, c \in Channels$), и

- если α имеет вид $c?e$ или $e := e'$, то при переходе от s к s' данные множества не изменяются, и
- если α имеет вид $c!e$, то при переходе от s к s' может измениться лишь множество $[c]^s$ путем добавления к нему терма e^s,

поэтому из (54) следует импликация $s \models E \perp_{\mathbf{C}} P_\dagger \Rightarrow s' \models E \perp_{\mathbf{C}} P_\dagger$.

Импликация $s \models E' \subseteq [c] \subseteq E'' \Rightarrow s' \models E'_{c,\alpha} \subseteq [c] \subseteq E''_{c,\alpha}$ следует из определения (53). ■

Теорема 8.

Пусть заданы

- РП $\mathcal{P}$, подмножество $E \subseteq \langle\mathcal{P}\rangle_0$, вершина $at \in G_{\mathcal{P}}$,
- множество $\{at_i \overset{\alpha_i}{\to} at \mid i \in I\}$ рёбер ГП $G_{\mathcal{P}}$ (с общим концом at), причем если $G_{\mathcal{P}}$ содержит ребро вида $at' \overset{\alpha}{\to} at$, не входящее в это множество, то вершина at' недостижима,
- множество $\{\varphi_i \mid i \in I\}$ формул, соответствующих вышеупомянутым ребрам, где $\forall\, i \in I \;\; at_i \models \varphi_i$, и φ_i состоит из следующих ЭФ:

$$\begin{cases} E \perp_{\mathbf{C}} P_\dagger \\ E'_{i,c} \subseteq [c] \subseteq E''_{i,c} \; (\forall\, c \in E_{\mathbf{C}}), \text{ где } E'_{i,c}, E''_{i,c} \subseteq Tm, \text{ и} \\ \text{равенства вида } e = e', \text{ где } e, e' \in Tm. \end{cases}$$

Тогда $at \models \varphi$, где φ состоит из следующих ЭФ:

$$\begin{cases} E \perp_{\mathbf{C}} P_\dagger \\ \bigcap_{i\in I}(E'_{i,c})_{c,\alpha_i} \subseteq [c] \subseteq \bigcup_{i\in I}(E''_{i,c})_{c,\alpha_i} \; (\forall\, c \in E_{\mathbf{C}}), \text{ и} \\ \text{равенства вида } e = e', \text{ где } e, e' \in Tm, \\ \qquad\qquad \text{входящие в каждую из формул } \varphi_i \; (i \in I). \end{cases}$$

Доказательство.
Данная теорема является следствием теорем 7 и 2. ■

Теорема 9.
Пусть заданы РП $\mathcal{P}$, и состояния $s, s' \in \Sigma_{\mathcal{P}_\dagger}$, такие, что $s \xrightarrow{c?\hat{x}} s'$.
Тогда верна импликация $s \models \{[c'] = \{e\}, c = c'\} \Rightarrow s' \models \{x = e\}$.

Доказательство.
Данная теорема непосредственно вытекает из определения выполнения действия вида $c?\hat{x}$ (см. пункт 2.2.5). ■

3.1.3. Редукция графов переходов

Если анализируемые свойства ГП $G_{\mathcal{P}}$ связаны только с его достижимыми вершинами, то при решении задач анализа таких свойств недостижимые вершины и связанные с ними ребра м.б. удалены из этого ГП. Будем называть такую операцию удаления **редукцией** ГП. Получившийся после такого удаления граф будем называть **редуцированным** ГП, и обозначать его той же записью $G_{\mathcal{P}}$. Если в редуцированном ГП обнаружатся неудаленные недостижимые вершины, то этот ГП можно еще раз редуцировать, и т.д.

Нахождение нереализуемых ребер и недостижимых вершин ГП можно производить с помощью следующих теорем.

Теорема 10.
Пусть задан РП $\mathcal{P}$. Если вершина $at \in G_{\mathcal{P}}$ такова, что

- $at \models \{[c] = \emptyset, c = c'\}$, где $c, c' \in Channels$, или
- $at \models \{k^{-1}[\circ] = \emptyset, k = k'\}$, где $k, k' \in Keys$,

и из at выходит ребро с меткой $c'?e$ или $?k'(e)$ соответственно, то это ребро нереализуемо. ■

Теорема 11.
Пусть задан РП $\mathcal{P}$. $\forall\, at \in G_{\mathcal{P}}$ верны следующие утверждения:

- если все рёбра с концом at нереализуемы, то at недостижима, и
- если at недостижима, то все рёбра с началом at нереализуемы. ■

3.1.4. Верификация распределенного процесса $\mathcal{P}_1$

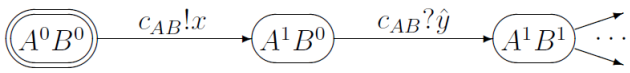
Применим доказанные выше теоремы для доказательства свойства (52) РП $\mathcal{P}_1$, описываемого схемой (41). ГП $G_{\mathcal{P}_1}$ имеет вид (47).

Теорема 10 и первое соотношение в (50), которое имеет вид

$$A^0B^0 \models \{\varphi_1, [c_{AB}] = \emptyset\} \tag{55}$$

обосновывают нереализуемость отмеченного черным кружочком ребра в ГП (47). По теореме 11, отсюда следует недостижимость вершины A^0B^1.

После редукции ГП (47) путем удаления недостижимых вершин и связанных с ними ребер получаем граф



(56)

В (56) существует единственная вершина (A^1B^1), удовлетворяющая условию в (52). Таким образом, требуется доказать, что $A^1B^1 \models x = y$.

Из (55) по теореме 8 следует, что $A^1B^0 \models \{\varphi_1, [c_{AB}] = \{x\}\}$, откуда по теоремам 8 и 9 следует желаемое свойство $A^1B^1 \models x = y$.

3.1.5. Верификация распределенного процесса $\mathcal{P}_3$

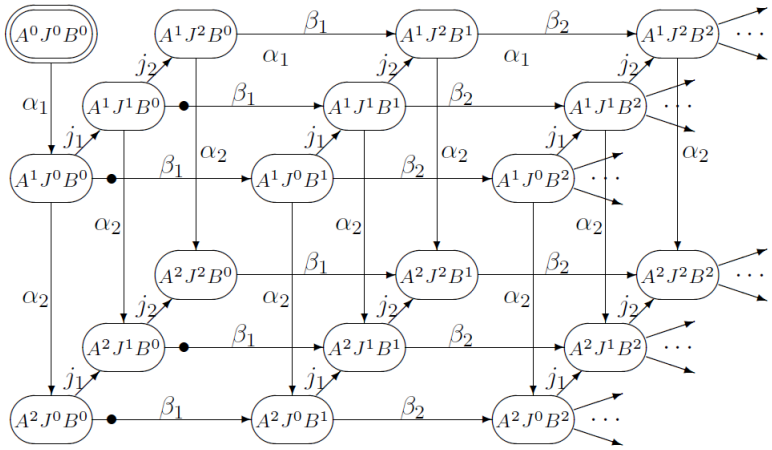
Рассмотрим теперь задачу доказательства свойства (52) для РП $\mathcal{P}_3$, описываемого схемой (44), в которой действия определяются согласно (43). ГП $G_{\mathcal{P}_3}$ имеет вид (49).

Теорема 10 и третье соотношение в (50), которое имеет вид

$$A^0J^0B^0 \models \{\varphi_3, [c_{AJ}] = \emptyset, [c_{BJ}] = \emptyset, [\bar{c}] = \emptyset\} \tag{57}$$

обосновывают нереализуемость отмеченных черными кружочками ребер в ГП (49). По теореме 11, отсюда следует недостижимость всех вершин верхнего яруса в ГП (49) за исключением вершины $A^0J^0B^0$.

После редукции ГП (49) путем удаления недостижимых вершин верхнего яруса и связанных с ними ребер получаем редуцированный ГП



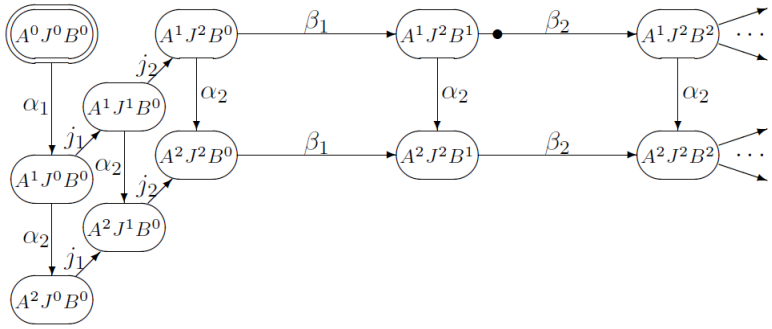
(58)

Далее мы приводим список соотношений, каждое из которых следует из предыдущих (первое следует из (57)) по теоремам 8 и 9:

$$\begin{array}{l} A^1J^0B^0 \models \{\varphi_3, [c_{AJ}] = \{\bar{c}\}, [c_{BJ}] = \emptyset, [\bar{c}] = \emptyset\} \\ A^2J^0B^0 \models \{\varphi_3, [c_{AJ}] = \{\bar{c}\}, [c_{BJ}] = \emptyset, [\bar{c}] = \{x\}\} \\ A^1J^1B^0 \models \{\varphi_3, [c_{AJ}] = \{\bar{c}\}, [c_{BJ}] = \emptyset, [\bar{c}] = \emptyset, u = \bar{c}\} \\ A^2J^1B^0 \models \{\varphi_3, [c_{AJ}] = \{\bar{c}\}, [c_{BJ}] = \emptyset, [\bar{c}] = \{x\}, u = \bar{c}\} \end{array} \tag{59}$$

По теореме 10, из данных соотношений следует нереализуемость отмеченных черными кружочками ребер в ГП (59). Удаляя эти ребра и соответствующие недостижимые вершины (используя теорему 11), получаем редуцирован-

ный ГП

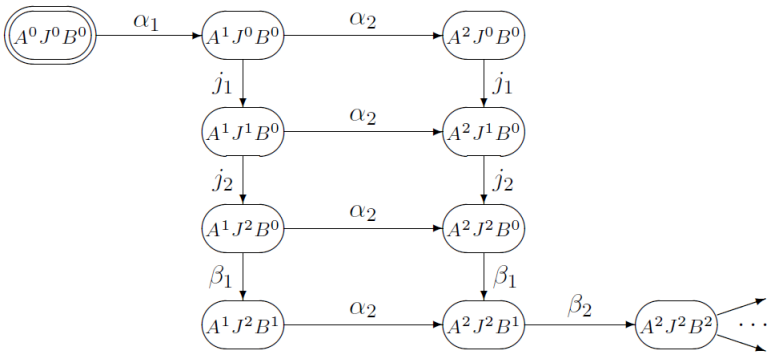


(60)

Из последнего соотношения в (59) при помощи теорем 8 и 9 получаем:

$$\begin{array}{l} A^1J^2B^0 \models \{\varphi_3, [c_{AJ}] = \{\bar{c}\}, [c_{BJ}] = \{u\}, [\bar{c}] = \emptyset, u = \bar{c}\} \\ A^1J^2B^1 \models \{\varphi_3, [c_{AJ}] = \{\bar{c}\}, [c_{BJ}] = \{u\}, [\bar{c}] = \emptyset, u = \bar{c}, v = u\} \end{array} \tag{61}$$

По теореме 10, из последнего соотношения в (61) следует нереализуемость отмеченного черным кружочком ребра в ГП (60). Удаляя это ребро и соответствующие недостижимые вершины (для нахождения которых используем теорему 11), получаем редуцированный ГП



(62)

Применяя теоремы 8 и 9, вычисляем формулы, соответствующие оставшимся вершинам:

$$\begin{array}{l} A^2J^2B^0 \models \{\varphi_3, [c_{AJ}] = \{\bar{c}\}, [c_{BJ}] = \{u\}, [\bar{c}] = \{x\}, u = \bar{c}\} \\ A^2J^2B^1 \models \{\varphi_3, [c_{AJ}] = \{\bar{c}\}, [c_{BJ}] = \{u\}, [\bar{c}] = \{x\}, u = \bar{c}, v = u\} \\ A^2J^2B^2 \models \{x = y\} \end{array} \tag{63}$$

Поскольку

- в ГП (62) вершина $A^2J^2B^2$ является единственной вершиной, удовлетворяющей условию в (52), и

- согласно последнему соотношению в (63), для этой вершины верно утверждение в (52),

то задача доказательства свойства (52) для РП $\mathcal{P}_3$ решена.

3.1.6. Теоремы, используемые для верификации распределенных процессов с защищенными ключами

В этом пункте будем использовать следующее обозначение:

$$\begin{array}{l}\forall\, E \subseteq Tm, \forall\, \alpha \in Act, \forall\, k \in Keys \\ E_{k,\alpha} = E \cup \{e\},\ \text{если}\ \alpha = !k(e),\ \text{и}\ E_{k,\alpha} = E,\ \text{иначе.}\end{array} \tag{64}$$

Теорема 12.

Пусть заданы РП $\mathcal{P}$, где $Var(\mathcal{P})_{\mathbf{C}} = \{\circ\}$, подмножество $E \subseteq \langle\mathcal{P}\rangle_0$, состояния $s, s' \in \Sigma_{\mathcal{P}_\dagger}$, такие, что $s \xrightarrow{\alpha_P} s'$, где $P \in \mathcal{P}$, и если $\alpha = !e$, то

$$\forall\, x \in E_{\mathbf{X}}\ \ x \perp_{\mathbf{K},E} e^s. \tag{65}$$

Тогда $\forall\, E', E'' \subseteq Tm$, $\forall\, k \in E_{\mathbf{K}} : k \neq public\_key(\ldots)$ верна импликация

$$s \models \left\{\begin{array}{l} E \perp_{\mathbf{K}} P_\dagger, k^{-1}[P_\dagger] \subseteq k^{-1}[\circ] \\ E' \subseteq k^{-1}[\circ] \subseteq E'' \end{array}\right\} \Rightarrow s' \models \left\{\begin{array}{l} E \perp_{\mathbf{K}} P_\dagger, k^{-1}[P_\dagger] \subseteq k^{-1}[\circ] \\ E'_{k,\alpha} \subseteq k^{-1}[\circ] \subseteq E''_{k,\alpha} \end{array}\right\}$$

Доказательство.

Значения формул $E \perp_{\mathbf{K}} P_\dagger$ и $k^{-1}[P_\dagger] \subseteq k^{-1}[\circ]$ в s зависит только от множеств $[P_\dagger]^s$ и $[\circ]_s$ (для $E \perp_{\mathbf{K}} P_\dagger$ это верно согласно (5)), и

- если α имеет вид $?e$ или $e := e'$, то при переходе от s к s' данные множества не изменяются, и
- если α имеет вид $!e$, то при переходе от s к s' может измениться лишь множество $[\circ]_s$ путем добавления к нему терма e^s,

поэтому из (65) следует импликация

$$s \models \{E \perp_{\mathbf{K}} P_\dagger, k^{-1}[P_\dagger] \subseteq k^{-1}[\circ]\} \Rightarrow s' \models \{E \perp_{\mathbf{K}} P_\dagger, k^{-1}[P_\dagger] \subseteq k^{-1}[\circ]\}.$$

Импликация $s \models E' \subseteq k^{-1}[\circ] \subseteq E'' \Rightarrow s' \models E'_{k,\alpha} \subseteq k^{-1}[\circ] \subseteq E''_{k,\alpha}$ следует из определения (64). ■

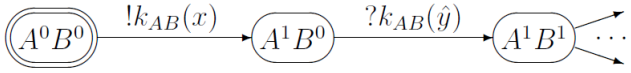
Кроме того, верны аналоги теоремы 8, с заменой

- $E \perp_{\mathbf{C}} P_\dagger$ на $\{E \perp_{\mathbf{K}} P_\dagger, k^{-1}[P_\dagger] \subseteq k^{-1}[\circ]\}$,
- $[c]$ на $k^{-1}[\circ]$, $E_{i,c}$ на $E_{i,k}$, $(E_{i,c})_{c,\alpha_i}$ на $(E_{i,k})_{k,\alpha_i}$,

и теоремы 9, с заменой $c?\hat{x}$ на $?k(\hat{x})$, $[c']$ на $(k')^{-1}[\circ]$, $c = c'$ на $k = k'$.

3.1.7. Верификация распределенного процесса $\mathcal{P}_2$

Доказательство свойства (52) для РП $\mathcal{P}_2$, описываемого схемой (42), проводится аналогично доказательству этого свойства для РП $\mathcal{P}_1$ в пункте 3.1.4. ГП $G_{\mathcal{P}_2}$ имеет вид (48). Так же обосновывается нереализуемость отмеченного черным кружочком ребра в ГП (48). По теореме 11, отсюда следует недостижимость вершины A^0B^1. После редукции ГП (48) получаем граф



(66)

В (66) существует единственная вершина (A^1B^1), удовлетворяющая условию в (52). Таким образом, требуется доказать, что $A^1B^1 \models x = y$. Это свойство следует из соотношения $A^1B^0 \models \{\varphi_2, k_{AB}^{-1}[\circ] = \{x\}\}$.

3.1.8. Верификация распределенного процесса $\mathcal{P}_4$

Доказательство свойства (52) для РП $\mathcal{P}_4$, описываемого схемой (44), где действия α_i, β_i, j_i $(i = 1, 2)$ определяются согласно (45), проводится аналогично доказательству этого свойства для РП $\mathcal{P}_3$ в пункте 3.1.5. ГП $G_{\mathcal{P}_4}$ имеет вид (49). Так же обосновывается нереализуемость отмеченных черными кружочками ребер в ГП (49). После редукции ГП (49) получаем такие же ГП (58), (60), (62), как и в случае верификации РП $\mathcal{P}_3$. Мы не будем излагать детально решение задачи верификации РП $\mathcal{P}_4$, приведем лишь соотношения, связанные с вершинами ГП (62) для данного случая.

$$
\begin{aligned}
&A^1J^0B^0 \models \{\varphi_4, k_{AJ}^{-1}[\circ] = \{\bar{k}\}, k_{BJ}^{-1}[\circ] = \emptyset, \bar{k}^{-1}[\circ] = \emptyset\}\\
&A^2J^0B^0 \models \{\varphi_4, k_{AJ}^{-1}[\circ] = \{\bar{k}\}, k_{BJ}^{-1}[\circ] = \emptyset, \bar{k}^{-1}[\circ] = \{x\}\}\\
&A^1J^1B^0 \models \{\varphi_4, k_{AJ}^{-1}[\circ] = \{\bar{k}\}, k_{BJ}^{-1}[\circ] = \emptyset, \bar{k}^{-1}[\circ] = \emptyset, u = \bar{k}\}\\
&A^2J^1B^0 \models \{\varphi_4, k_{AJ}^{-1}[\circ] = \{\bar{k}\}, k_{BJ}^{-1}[\circ] = \emptyset, \bar{k}^{-1}[\circ] = \{x\}, u = \bar{k}\}\\
&A^1J^2B^0 \models \{\varphi_4, k_{AJ}^{-1}[\circ] = \{\bar{k}\}, k_{BJ}^{-1}[\circ] = \{u\}, \bar{k}^{-1}[\circ] = \emptyset, u = \bar{k}\}\\
&A^1J^2B^1 \models \{\varphi_4, k_{AJ}^{-1}[\circ] = \{\bar{k}\}, k_{BJ}^{-1}[\circ] = \{u\}, \bar{k}^{-1}[\circ] = \emptyset, u = \bar{k}, v = u\}\\
&A^2J^2B^0 \models \{\varphi_4, k_{AJ}^{-1}[\circ] = \{\bar{k}\}, k_{BJ}^{-1}[\circ] = \{u\}, \bar{k}^{-1}[\circ] = \{x\}, u = \bar{k}\}\\
&A^2J^2B^1 \models \{\varphi_4, k_{AJ}^{-1}[\circ] = \{\bar{k}\}, k_{BJ}^{-1}[\circ] = \{u\}, \bar{k}^{-1}[\circ] = \{x\}, u = \bar{k}, v = u\}\\
&A^2J^2B^2 \models \{x = y\}.
\end{aligned}
$$

3.2. Верификация протокола Yahalom

В этом и следующем параграфе рассматривается другой метод верификации КП, основанный на использовании теоремы 6. Данный метод не описывается явно, т.к. его содержание можно понять по приводимым ниже примерам его применения в задаче верификации КП аутентификации Yahalom (в этом параграфе) и КП передачи ШС между несколькими агентами (в параграфе 3.3).

3.2.1. Описание протокола Yahalom

КП Yahalom предназначен для аутентификации (т.е. проверки подлинности) агентов, взаимодействующих по открытому каналу $\circ$, и передачи сеансовых ключей между этими агентами.

Предполагается что

- заданы множество агентов Ag, а также агент J, называемый **доверенным посредником**, данные агенты могут взаимодействовать друг с другом по открытому каналу $\circ$,
- каждый агент $A \in Ag$ имеет общий секретный ключ k_{AJ} с доверенным посредником J, на котором A и J могут шифровать и дешифровать сообщения, используя симметричную систему шифрования, причем только A и J знают ключ k_{AJ}.

В каждом сеансе КП Yahalom принимают участие следующие агенты: **инициатор** $A \in Ag$, доверенный посредник J, и **респондер** $B \in Ag$. Каждый агент из Ag в одних сеансах м.б. инициатором, а в других – респондером. Один и тот же агент м.б. и инициатором и респондером в одном и том же сеансе (т.е. возможно, что $A = B$). Выполнение сеанса КП Yahalom с инициатором A, респондером B и доверенным посредником J представляет собой совокупность четырех пересылок сообщений:

$$\begin{array}{lcl} 1. \quad A \to B & : & A, n_A \\ 2. \quad B \to J & : & B, k_{BJ}(A, n_A, n_B) \\ 3. \quad J \to A & : & k_{AJ}(B, k, n_A, n_B), k_{BJ}(A, k) \\ 4. \quad A \to B & : & k_{BJ}(A, k), k(n_B) \end{array} \tag{67}$$

Пересылки в (67) имеют следующий смысл:

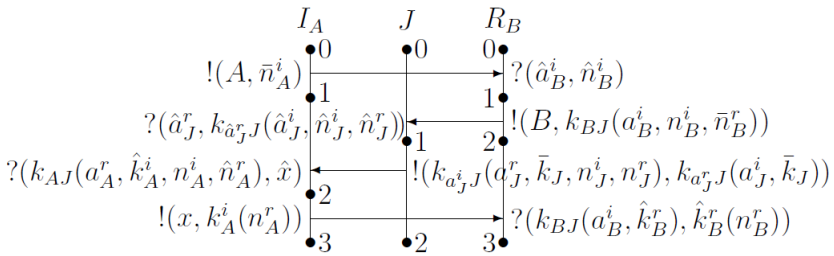
1) A посылает B запрос на аутентификацию и генерацию сеансового ключа k, этот запрос состоит из имени агента A и нонса n_A,

2) B посылает J запрос на генерацию сеансового ключа k, в свой запрос он включает своё имя, имя агента A, для связи с которым нужен этот ключ, полученный нонс n_A, и свой нонс n_B,

3) J генерирует сеансовый ключ k и посылает A пару сообщений,
   - из первого сообщения A может извлечь сеансовый ключ k,
   - а второе предназначено для того, чтобы A переслал его B,

4) A посылает B пару сообщений,
   - первое из которых было получено им от J, агент B может извлечь из этого сообщения сеансовый ключ k, и
   - используя ключ k, агент B дешифрует второе сообщение, если результат дешифрования совпадает с его нонсом n_B, то это является для него доказательством того, что отправителем этого сообщения был именно A.

Формальное описание сеанса КП Yahalom изображается схемой



(68)

В этой схеме левая и правая нити соответствуют ПП I_A и R_B, описывающим поведение инициатора A и респондера B соответственно, средняя нить соответствует ПП, описывающему поведение посредника J, этот ПП обозначается тем же символом J. Смысл переменных в этих ПП усматривается из сопоставления действий в этих ПП с соответствующими действиями в (67). Верхний индекс i или r при какой-либо переменной означает, что она предположительно содержит информацию об инициаторе (i) или респондере (r) данного сеанса.

Предполагаем, что $Agent(I_A) = A$, $Agent(R_B) = B$, $Agent(J) = J$.

РП $\mathcal{P}$, соответствующий КП Yahalom, имеет вид

$$\mathcal{P} = \{\{I^*_A \mid A \in Ag\}, \{R^*_B \mid B \in Ag\}, J^*\}. \tag{69}$$

Ниже мы будем использовать следующие обозначения:

- если $\mathcal{P}$ – РП, и π – путь в $\Sigma_{\mathcal{P}_\dagger}$, то запись $\pi \ni P^{i,i'} : s \overset{\alpha}{\to} s'$ означает, что π содержит ребро $s \overset{\alpha_P}{\to} s'$, и $at_{s_P} = i$, $at_{s'_P} = i'$,
- запись $s \models E \perp_{\mathbf{K}} e$ обозначает утверждение $\forall\, x \in E_{\mathbf{X}} \;\; x \perp_{\mathbf{K},E} e^s$.

Нетрудно доказать, что

$$s \models E \perp_{\mathbf{K}} (e, e') \Leftrightarrow s \models E \perp_{\mathbf{K}} e \quad \text{и} \quad s \models E \perp_{\mathbf{K}} e'. \tag{70}$$

3.2.2. Свойства протокола Yahalom

Свойства РП (69), которые будут верифицированы:

- **секретность** ключей и нонсов n^r_B:

$$\forall\, s \in \Sigma_{\mathcal{P}_\dagger} \quad s \models E \perp_{\mathbf{K}} P_\dagger, \quad \text{где } E = \{k_{BJ}, k_J, n^r_B \mid B \in Ag\} \tag{71}$$

- **аутентификация инициатора перед респондером**: для любых $R_B \in \mathcal{P}$, $s \in \Sigma_{\mathcal{P}_\dagger}$, если $s \models at_{R_B} = 3$, то $\exists\, I_A \in \mathcal{P}$:

$$s \models \{at_{I_A} = 3, a^r_A = B, a^i_B = A, n^i_A = n^i_B, n^r_A = n^r_B, k^i_A = k^r_B\}, \tag{72}$$

- **аутентификация респондера перед инициатором**: для любых $I_A \in \mathcal{P}$, $s \in \Sigma_{\mathcal{P}_\dagger}$, если $s \models at_{I_A} = 2$, то $\exists\, R_B \in \mathcal{P}$:

$$s \models \{at_{R_B} = 2, a^r_A = B, a^i_B = A, n^i_A = n^i_B, n^r_A = n^r_B\}. \tag{73}$$

3.2.3. Секретность ключей и нонсов n^r_B

Докажем (71) от противного.

Предположим, что $S = \{s \in \Sigma_{\mathcal{P}_\dagger} \mid s \not\models \varphi\} \neq \emptyset$, где φ – формула в (71).

$\forall\, s \in S$ обозначим записью π_s путь минимальной длины из 0 в s. Пусть s – состояние из S с наименьшей длиной π_s. Т.к. $0 \models \varphi$, то $s \neq 0$.

Пусть $s' \overset{\alpha_P}{\to} s$ – ребро из π_s с концом в s.

Из определения s следует, что $s' \models \varphi$, $s \not\models \varphi$. Если бы было верно $P = P_\dagger$, то из теоремы 3 следует, что $s \models \varphi$, т.е. имеем противоречие.

Поэтому $P \in \{I_A, R_B, J \mid A, B \in Ag\}$, и

$$\alpha_P = !e,\ [\circ]_s = [\circ]_{s'} \cup \{e\},\ \exists\, y \in E_{\mathbf{X}} : \neg(y \perp_{\mathbf{K},E} e^s). \tag{74}$$

Перебором всех вариантов обоснования существования ребра $s' \overset{\alpha_P}{\to} s$ со свойствами (74) находим единственное возможное обоснование:

$$\pi_s \ni I^{2,3}_A : s' \overset{!e}{\to} s, \quad \text{где } e = (x, k^i_A(n^r_A)). \tag{75}$$

Т.к. $s' \models at_{I_A} = 2$, то $\exists\, s_1 \leq_{\pi_s} s'$:

$$\pi_s \ni I^{1,2}_A : s'_1 \overset{?e_1}{\to} s_1, \text{ где } e_1 = (k_{AJ}(a^r_A, \hat{k}^i_A, n^i_A, \hat{n}^r_A), \hat{x}). \tag{76}$$

Т.к. $s_1 \leq_{\pi_s} s'$ и $s' \models \varphi$, то $s_1 \models \varphi$. В частности, $s_1 \models E \perp_{\mathbf{K}} e_1$. По (70), отсюда следует, что $s_1 \models E \perp_{\mathbf{K}} x$.

По теореме 6, из $s_1 \models \varphi$, $e^s_1 \in [\circ]_{s_1}$ и $k_{AJ} \in E$ следует, что $\exists\, s_2 \leq_{\pi_s} s'_1$: π_s содержит ребро $s'_2 \overset{(!e_2)_P}{\longrightarrow} s_2$, где $P \in \mathcal{P}$ и первая компонента $k_{AJ}(\ldots)$ терма e^s_1 входит в e^s_2. Перебором всех вариантов обоснования существования ребра с такими свойствами находим единственное обоснование:

$$\begin{cases} \pi_s \ni J^{1,2} : s'_2 \overset{!e_2}{\to} s_2, \text{ где } e_2 = (k_{a^i_J J}(a^r_J, \bar{k}_J, n^i_J, n^r_J), \ldots) \\ k_{(a^i_J)^s J}((a^r_J)^s, \bar{k}_J, \ldots) = k_{AJ}(a^r_A, (k^i_A)^s, \ldots) \end{cases} \tag{77}$$

(многоточие в (77) и ниже обозначает компоненту пары, не представляющую интерес для рассмотрения). Из (77) следует, что $(k^i_A)^s = \bar{k}_J$, поэтому $s \models E \perp_{\mathbf{K}} k^i_A(n^r_A)$. Учитывая установленное выше свойство $s_1 \models E \perp_{\mathbf{K}} x$, на основании (70) получаем: $s \models E \perp_{\mathbf{K}} (x, k^i_A(n^r_A))$, т.е. $s \models E \perp_{\mathbf{K}} e$, что противоречит предположению $s \not\models E \perp_{\mathbf{K}} e$. ■

Доказанное свойство $\forall\, s \in \Sigma_{\mathcal{P}_\dagger}\ \ s \models \varphi$ будет использоваться ниже.

В излагаемых ниже доказательствах при каждом применении теоремы 6 имеется единственный вариант обоснования существования ребра (35) в графе $\Sigma_{\mathcal{P}_\dagger}$, и мы будем сразу будем излагать это обоснование, без упоминания о единственности варианта такого обоснования. Эта единственность обеспечивается подходящим определением действий вида $!e$ в ПП, входящих в рассматриваемый РП.

3.2.4. Аутентификация инициатора перед респондером

Пусть ПП $R_B \in \mathcal{P}$ и состояние $s \in \Sigma_{\mathcal{P}_\dagger}$ таковы, что $s \models at_{R_B} = 3$.

Докажем, что $\exists\, I_A \in \mathcal{P}$: выполнено (72).

Пусть π – путь из 0 в s. Из $s \models at_{R_B} = 3$ следует, что $\exists\, s_1 \leq_\pi s$:

$$\pi \ni R_B^{2,3} : s_1' \overset{?e_1}{\to} s_1, \text{ где } e_1 = (k_{BJ}(a_B^i, \hat{k}_B^r), \hat{k}_B^r(n_B^r)).$$

По теореме 6, из $s_1 \models \varphi$, $e_1^s \in [\circ]_{s_1}$ и $k_{BJ} \in E$ следует, что $\exists\, s_2 \leq_\pi s_1'$:

$$\begin{cases} \pi \ni J^{1,2} : s_2' \overset{!e_2}{\to} s_2, \text{ где } e_2 = (\ldots, k_{a_J^r J}(a_J^i, \bar{k}_J)) \\ k_{(a_J^r)^s J}((a_J^i)^s, \bar{k}_J) = k_{BJ}((a_B^i)^s, (k_B^r)^s) \end{cases} \tag{78}$$

Из второго равенства в (78) следует, что

$$(a_J^r)^s = B, (a_J^i)^s = (a_B^i)^s, \bar{k}_J = (k_B^r)^s. \tag{79}$$

Из $s_2' \models at_J = 1$ следует, что $\exists\, s_3 \leq_\pi s_2'$:

$$\pi \ni J^{0,1} : s_3' \overset{?e_3}{\to} s_3, \text{ где } e_3 = (\ldots, k_{\hat{a}_J^r J}(\hat{a}_J^i, \hat{n}_J^i, \hat{n}_J^r)). \tag{80}$$

Из (79) и (80) следует, что $k_{BJ}(3 \text{ терма}) \subseteq e_3^s \in [\circ]_{s_3}$, откуда по теореме 6, с учетом $s_3 \models \varphi$ и $k_{BJ} \in E$ получаем: $\exists\, s_4 \leq_\pi s_3'$:

$$\begin{cases} \pi \ni R_{\grave{B}}^{1,2} : s_4' \overset{!e_4}{\to} s_4, \text{ где } e_4 = (\ldots, k_{\grave{B}J}(a_{\grave{B}}^i, n_{\grave{B}}^i, \bar{n}_{\grave{B}}^r)) \\ k_{\grave{B}J}((a_{\grave{B}}^i)^s, (n_{\grave{B}}^i)^s, \bar{n}_{\grave{B}}^r) = k_{BJ}((a_B^i)^s, (n_J^i)^s, (n_J^r)^s) \end{cases} \tag{81}$$

Из второго равенства в (81) следует, что

$$\grave{B} = B, (n_B^i)^s = (n_J^i)^s, \bar{n}_B^r = (n_J^r)^s. \tag{82}$$

По теореме 6, из $s_1 \models \varphi$, $(k_B^r(n_B^r))^s \subseteq e_1^s \in [\circ]_{s_1}$, и $(k_B^r)^s = \bar{k}_J \in E$ следует, что $\exists\, s_5 \leq_\pi s_1'$:

$$\begin{cases} \pi \ni I_A^{2,3} : s_5' \overset{!e_5}{\to} s_5, \text{ где } e_5 = (\ldots, k_A^i(n_A^r)) \\ (k_A^i(n_A^r))^s = \bar{k}_J(n_B^r) \end{cases} \tag{83}$$

Из второго равенства в (83) следует, что

$$(k_A^i)^s = \bar{k}_J, (n_A^r)^s = n_B^r. \tag{84}$$

Из $s_5 \models at_{I_A} = 2$ следует, что $\exists\, s_6 \leq_\pi s_5'$:

$$\pi \ni I_A^{1,2} : s_6' \overset{?e_6}{\to} s_6, \text{ где } e_6 = (k_{AJ}(a_A^r, \hat{k}_A^i, n_A^i, \hat{n}_A^r), \ldots). \tag{85}$$

Из (84) и (85) следует, что

$$k_{AJ}(a_A^r,(k_A^i)^s,n_A^i,(n_A^r)^s)=k_{AJ}(a_A^r,\bar{k}_J,n_A^i,n_B^r)\subseteq e_6^s\in[\circ]_{s_6}. \tag{86}$$

По теореме 6, из $s_6\models\varphi$, $k_{AJ}\in E$, и (86) следует, что $\exists\, s_7\leq_\pi s_6'$:

$$\left\{\begin{array}{l}\pi\ni \grave{J}^{1,2}:s_7'\overset{!e_7}{\to}s_7,\ \text{где } e_7=(k_{a^i_{\grave{J}}\grave{J}}(a^r_{\grave{J}},\bar{k}_{\grave{J}},n^i_{\grave{J}},n^r_{\grave{J}}),\ldots)\\ k_{(a^i_{\grave{J}})^sJ}((a^r_{\grave{J}})^s,\bar{k}_{\grave{J}},(n^i_{\grave{J}})^s,(n^r_{\grave{J}})^s)=k_{AJ}(a_A^r,\bar{k}_J,n_A^i,n_B^r)\end{array}\right. \tag{87}$$

Из второго равенства в (87) следует, что

$$(a^i_{\grave{J}})^s=A,(a^r_{\grave{J}})^s=a_A^r,\grave{J}=J,(n_J^i)^s=n_A^i,(n_J^r)^s=\bar{n}_B^r. \tag{88}$$

Свойство (72) следует из (79), (82), (84), (88). ■

3.2.5. Аутентификация респондера перед инициатором

Пусть ПП $I_A\in\mathcal{P}$ и состояние $s\in\Sigma_{\mathcal{P}_\dagger}$ таковы, что $s\models at_{I_A}=2$.

Докажем, что $\exists\, R_B\in\mathcal{P}$: выполнено (73).

Пусть π – путь из 0 в s. Из $s\models at_{I_A}=2$ следует, что $\exists\, s_1\leq_\pi s$:

$$\pi\ni I_A^{1,2}:s_1'\overset{?e_1}{\to}s_1,\ \text{где } e_1=(k_{AJ}(a_A^r,\hat{k}_A^i,n_A^i,\hat{n}_A^r),\ldots). \tag{89}$$

По теореме 6, из $s_1\models\varphi$, $k_{AJ}(4\text{ терма})\subseteq e_1^s\in[\circ]_{s_1}$ и $k_{AJ}\in E$, следует, что $\exists\, s_2\leq_\pi s_1'$:

$$\left\{\begin{array}{l}\pi\ni J^{1,2}:s_2'\overset{!e_2}{\to}s_2,\ \text{где } e_2=(k_{a^i_JJ}(a_J^r,\bar{k}_J,n_J^i,n_J^r),\ldots)\\ k_{(a^i_J)^sJ}((a_J^r)^s,\bar{k}_J,(n_J^i)^s,(n_J^r)^s)=k_{AJ}(a_A^r,(k_A^i)^s,n_A^i,(n_A^r)^s)\end{array}\right. \tag{90}$$

Из второго равенства в (90) следует, что

$$(a_J^i)^s=A,(a_J^r)^s=a_A^r,\bar{k}_J=(k_A^i)^s,(n_J^i)^s=n_A^i,(n_J^r)^s=(n_A^r)^s. \tag{91}$$

Из $s_2\models at_J=1$ следует, что $\exists\, s_3\leq_\pi s_2'$:

$$\pi\ni J^{0,1}:s_3'\overset{?e_3}{\to}s_3,\ \text{где } e_3=(\ldots,k_{\hat{a}_J^rJ}(\hat{a}_J^i,\hat{n}_J^i,\hat{n}_J^r)). \tag{92}$$

Из (91) и (92) следует, что $k_{a_A^rJ}(A,n_A^i,(n_A^r)^s)\subseteq e_3^s\in[\circ]_{s_3}$, откуда по теореме 6, учитывая $s_3\models\varphi$, и $k_{a_A^rJ}\in E$, получаем: $\exists\, s_4\leq_\pi s_3'$:

$$\left\{\begin{array}{l}\pi\ni R_B^{1,2}:s_4'\overset{!e_4}{\to}s_4,\ \text{где } e_4=(\ldots,k_{BJ}(a_B^i,n_B^i,\bar{n}_B^r))\\ k_{BJ}((a_B^i)^s,(n_B^i)^s,\bar{n}_B^r)=k_{a_A^rJ}(A,n_A^i,(n_A^r)^s).\end{array}\right. \tag{93}$$

Второе равенство в (93) влечёт равенства, из которых следует (73):

$$B=a_A^r,(a_B^i)^s=A,(n_B^i)^s=n_A^i,\bar{n}_B^r=(n_A^r)^s.\ \blacksquare$$

3.3. Верификация протокола передачи шифрованных сообщений между несколькими агентами

В этом пункте рассматривается пример верификации КП, предназначенного для передачи ШС по открытому каналу между несколькими агентами. Данный КП является обобщением рассмотренного в пункте 2.6.3 КП Wide-Mouth Prog, представляемого РП $\mathcal{P}_4$.

3.3.1. Описание протокола

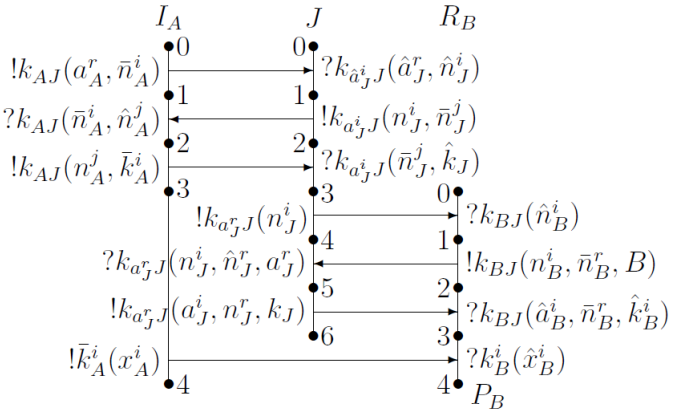
Участники этого протокола – агенты из множества $Ag \subseteq Agents$ и доверенный посредник J. Каждый агент $A \in Ag$ использует для связи с J ключ k_{AJ}, доступный только A и J. Сеанс передачи сообщения x в зашифрованном виде от агента $A \in Ag$ агенту $B \in Ag$ включает в себя следующие действия:

- обмен сообщениями между A и J, в результате чего J узнает имя A отправителя, имя B получателя, и ключ k, на котором будет зашифровано сообщение x от A для получателя B,
- обмен сообщениями между J и B, в результате чего B узнает имя A отправителя сообщения, которое B получит от A, и ключ k, на котором будет зашифровано это сообщение,
- пересылка ШС $k(x)$ от A к B.

Выполнение сеанса данного КП с инициатором A, респондером B и доверенным посредником J представляет собой следующую совокупность пересылок сообщений:

$$\begin{array}{lllll} 1. & A \to J & : & k_{AJ}(A, n_A) \\ 2. & J \to A & : & k_{AJ}(n_A, n_J) \\ 3. & A \to J & : & k_{AJ}(n_J, k) \\ 4. & J \to B & : & k_{BJ}(n_A) \\ 5. & B \to J & : & k_{BJ}(n_A, n_B, B) \\ 6. & J \to B & : & k_{BJ}(A, n_B, k) \\ 7. & A \to B & : & k(x) \end{array} \tag{94}$$

Данный сеанс представляется следующей схемой:



(95)

РП $\mathcal{P}$, соответствующий этому КП, имеет вид (69).
Свойства этого КП, которые должны быть верифицированы:

- **секретность** ключей, передаваемых сообщений и нонсов:

$$\forall\, s \in \Sigma_{\mathcal{P}_\dagger}\ \ s \models E \perp_{\mathbf{K}} P_\dagger,\ \ \text{где } E = \{k_{AJ}, k_A^i, x_A^i, n_A^i \mid A \in Ag\} \tag{96}$$

- **целостность** передаваемых сообщений:

$$\begin{array}{l}\forall\, R_B \in \mathcal{P},\, \forall\, s \in \Sigma_{\mathcal{P}_\dagger}, \text{ если } s \models at_{R_B} = 4, \text{ то } \exists\ I_A \in \mathcal{P}: \\ s \models \{at_{I_A} = 4, a_A^r = B, a_B^i = A, n_A^i = n_B^i, k_A^i = k_B^i, x_A^i = x_B^i\}\end{array} \tag{97}$$

3.3.2. Верификация протокола

Доказательство свойства секретности (96) дословно повторяет начало рассуждений в пункте 3.2.3 по доказательству аналогичного свойства протокола Yahalom, с тем лишь отличием что не существует ни одного варианта обоснования существования ребра $s' \overset{\alpha_P}{\to} s$ со свойствами (74).

Докажем свойство целостности (97). Будем использовать в этом доказательстве доказанное выше свойство (96) (не упоминая об этом).

Пусть ПП $R_B \in \mathcal{P}$ и состояние $s \in \Sigma_{\mathcal{P}_\dagger}$ таковы, что $s \models at_{R_B} = 4$. Докажем, что $\exists\, I_A \in \mathcal{P}$: выполнено утверждение во второй строчке (97).

Пусть π – путь из 0 в s. Из $s \models at_{R_B} = 4$ следует, что

$$\begin{array}{l}\exists\, s_1 \le_\pi s:\ \pi \ni R_B^{3,4} : s_1' \overset{?e_1}{\to}\ s_1,\ \text{где } e_1 = k_B^i(\hat{x}_B^i) \\ \exists\, s_2 \le_\pi s_1':\ \pi \ni R_B^{2,3} : s_2' \overset{?e_2}{\to}\ s_2,\ \text{где } e_2 = k_{BJ}(\hat{a}_B^i, \bar{n}_B^r, \hat{k}_B^i)\end{array} \tag{98}$$

По теореме 6, из второй строки в (98), $e_2^s \in [\circ]_{s_2}$, $k_{BJ} \in E$, следует:

$$\left\{\begin{array}{l}\exists\, s_3 \le_\pi s_2' : \pi \ni J^{5,6} : s_3' \overset{!e_3}{\to}\ s_3,\ \text{где } e_3 = k_{a_J^r J}(a_J^i, n_J^r, k_J) \\ k_{(a_J^r)^s J}((a_J^i)^s, (n_J^r)^s, (k_J)^s) = k_{BJ}(\hat{a}_B^i, \bar{n}_B^r, \hat{k}_B^i)\end{array}\right. \tag{99}$$

Из второй строки в (99) следует, что

$$(a_J^r)^s = B,\ (a_J^i)^s = (a_B^i)^s,\ (n_J^r)^s = \bar{n}_B^r,\ (k_J)^s = (k_B^i)^s. \tag{100}$$

Из первой строки в (99), с учетом (100), получаем:

$$\exists\, s_4 \le_\pi s_3' : \pi \ni J^{4,5} : s_4' \overset{?e_4}{\to}\ s_4,\ \text{где } e_4 = k_{BJ}(n_J^i, n_B^r, B). \tag{101}$$

По теореме 6, из (101), и того, что $e_4^s \in [\circ]_{s_4}$, $k_{BJ} \in E$, следует:

$$\left\{\begin{array}{l}\exists\, s_5 \le_\pi s_4' : \pi \ni \grave{B}^{1,2} : s_5' \overset{!e_5}{\to}\ s_5,\ \text{где } e_5 = k_{\grave{B}J}(n_{\grave{B}}^i, \bar{n}_{\grave{B}}^r, \grave{B}) \\ k_{\grave{B}J}((n_{\grave{B}}^i)^s, \bar{n}_{\grave{B}}^r, \grave{B}) = k_{BJ}((n_J^i)^s, \bar{n}_B^r, B)\end{array}\right. \tag{102}$$

Из второй строки в (102) следует, что

$$\bar{n}^r_{\grave{B}} = \bar{n}^r_B,\ \grave{B} = B,\ (n^i_B)^s = (n^i_J)^s. \tag{103}$$

Из (101) следует, что

$$\exists\, s_6 \leq_\pi s'_4 : \pi \ni J^{2,3} : s'^{\,?e_6}_6 \to\ s_6,\ \text{где}\ e_6 = k_{a^i_J J}(n^j_J, k_J). \tag{104}$$

По теореме 6, из (104), и того, что $e^s_6 \in [\circ]_{s_6}$, $k_{(a^i_J)^s J} \in E$, следует:

$$\begin{cases} \exists\, s_7 \leq_\pi s'_6 : \pi \ni A^{2,3} : s'^{\,!e_7}_7 \to\ s_7,\ \text{где}\ e_7 = k_{AJ}(n^j_A, \bar{k}^i_A) \\ k_{AJ}((n^j_A)^s, \bar{k}^i_A) = k_{(a^i_J)^s J}(\bar{n}^j_J, (k_J)^s) \end{cases} \tag{105}$$

Из второй строки в (105) получаем:

$$A = (a^i_J)^s,\ (n^j_A)^s = \bar{n}^j_J,\ \bar{k}^i_A = (k_J)^s \tag{106}$$

Из первой строки в (105) получаем:

$$\exists\, s_8 \leq_\pi s'_7 : \pi \ni I^{1,2}_A : s'^{\,?e_8}_8 \to\ s_8,\ \text{где}\ e_8 = k_{AJ}(\bar{n}^i_A, \hat{n}^j_A). \tag{107}$$

По теореме 6, из (107), и того, что $e^s_8 \in [\circ]_{s_8}$, $k_{AJ} \in E$, следует:

$$\begin{cases} \exists\, s_9 \leq_\pi s'_8 : \pi \ni \grave{J}^{1,2} : s'^{\,!e_9}_9 \to\ s_9,\ \text{где}\ e_9 = k_{a^i_{\grave{J}} J}(n^i_{\grave{J}}, \bar{n}^j_{\grave{J}}) \\ k_{(a^i_{\grave{J}})^s J}((n^i_{\grave{J}})^s, \bar{n}^j_{\grave{J}}) = k_{AJ}(\bar{n}^i_A, (n^j_A)^s) \end{cases} \tag{108}$$

Из второй строки в (108) получаем:

$$(a^i_{\grave{J}})^s = A,\ (n^i_{\grave{J}})^s = \bar{n}^i_A,\ \bar{n}^j_{\grave{J}} = (n^j_A)^s \tag{109}$$

Из (106) и (109) получаем:

$$\bar{n}^j_{\grave{J}} = \bar{n}^j_J = (n^j_A)^s,\ \grave{J} = J,\ (a^i_J)^s = A,\ (n^i_J)^s = \bar{n}^i_A. \tag{110}$$

Из (100) и (106) получаем:

$$(k^i_B)^s = (k_J)^s = \bar{k}^i_A \in E, \tag{111}$$

поэтому по теореме 6, из первой строки в (98) и $e^s_1 \in [\circ]_{s_1}$ следует:

$$\begin{cases} \exists\, s_{11} \leq_\pi s'_1 : \pi \ni \grave{A}^{3,4} : s'^{\,!e_{11}}_{11} \to\ s_{11},\ \text{где}\ e_{11} = \bar{k}^i_{\grave{A}}(x^i_{\grave{A}}) \\ \bar{k}^i_{\grave{A}}(x^i_{\grave{A}}) = (k^i_B)^s((x^i_B)^s) \end{cases} \tag{112}$$

Из второй строки в (112) и (111) получаем:

$$\bar{k}^i_{\grave{A}} = (k^i_B)^s = \bar{k}^i_A,\ \grave{A} = A,\ x^i_A = (x^i_B)^s. \tag{113}$$

Из первой строки в (108) и (110) получаем:

$$\exists\, s_{10} \leq_\pi s'_9 : \pi \ni J^{0,1} : s'_{10} \overset{?e_{10}}{\rightarrow} s_{10},\ \text{где}\ e_{10} = k_{\hat{a}^i_J J}(\hat{a}^r_J, \hat{n}^i_J). \tag{114}$$

Из (100) и (110) следует, что $e^s_{10} = k_{AJ}(B, \bar{n}^i_A)$.
По теореме 6, из (114), $e^s_{10} \in [\circ]_{s_{10}}$, $k_{AJ} \in E$, следует:

$$\begin{cases} \exists\, s_{12} \leq_\pi s'_{10} : \pi \ni \grave{A}^{0,1} : s'_{12} \overset{!e_{12}}{\rightarrow} s_{12},\ \text{где}\ e_{12} = k_{\grave{A}J}(a^r_{\grave{A}}, \bar{n}^i_{\grave{A}}) \\ k_{\grave{A}J}(a^r_{\grave{A}}, \bar{n}^i_{\grave{A}}) = k_{AJ}(B, \bar{n}^i_A) \end{cases} \tag{115}$$

Из второй строки в (115) получаем:

$$\bar{n}^i_{\grave{A}} = \bar{n}^i_A,\ \grave{A} = A,\ a^r_A = B. \tag{116}$$

Утверждение (97) обосновывается следующим образом:

- $s \models at_{I_A} = 4$ следует из (112), (113): $s_{11} \models at_{\grave{A}} = 4$, $\grave{A} = A$, $s_{11} \leq_\pi s$,
- $s \models a^r_A = B$ следует из (116),
- $s \models a^i_B = A$ следует из (100) и (106),
- $s \models n^i_A = n^i_B$ следует из (103) и (110),
- $s \models k^i_A = k^i_B$ следует из (111),
- $s \models x^i_A = x^i_B$ следует из (113). ■

4. Заключение

В настоящей работе была построена новая модель КП, и показаны примеры ее использования для решения задач верификации свойств целостности, секретности и соответствия.

Для дальнейшей деятельности по развитию данной модели и основанных на ней методов верификации можно назвать следующие задачи:

- развитие языков спецификаций свойств КП, позволяющих выражать например свойства нулевого разглашения в КП аутентификации, свойства неотслеживаемости в КП электронных платежей, свойства анонимности и правильности подсчета голосов в КП электронного голосования, и разработка методов верификации свойств, выражаемых на этих языках,
- построение методов автоматизированного синтеза КП по описанию свойств, которым они должны удовлетворять.

Список литературы

[21CDS] Veronique Cortier, Stephanie Delaune, and Vaishnavi Sundararajan. A Decidable Class of Security Protocols for Both Reachability and Equivalence Properties. Journal of Automated Reasoning, 65:479–520, April 2021.

[21RCSSS] Roggenbach, M., Cerone, A., Schlingloff, H., Schneider, G., Shaikh, S.A., Formal verification of security protocols, in: Formal Methods for Software Engineering: Languages, Methods, Application Domains (Texts in Theoretical Computer Science. An EATCS Series) 1st ed., Springer International Publishing, 2021.

[17CW] Veronique Cortier and Cyrille Wiedling. A formal analysis of the Norwegian E-voting protocol. Journal of Computer Security, 25(15777):21–57, 2017.

[16ABF] M. Abadi, B. Blanchet, C. Fournet. The Applied Pi Calculus: Mobile Values, New Names, and Secure Communication. [Research Report] ArXiv. 2016, pp.110. `hal-01423924`, `https://arxiv.org/abs/1609.03003`

[16B] Bruno Blanchet, Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif, 2016.

[16YEMM] Fan Yang, Santiago Escobar, Catherine A Meadows, Jose Meseguer. Strand Spaces with Choice via a Process Algebra Semantics. PPDP '16: Proceedings of the 18th International Symposium on Principles and Practice of Declarative Programming, September 2016, pages 76–89.

[14CK] Veronique Cortier, Steve Kremer. Formal Models and Techniques for Analyzing Security Protocols: A Tutorial. Foundations and Trends in Programming Languages, 1(3):151–267, (2014)

[13LP] Yongjian Li, Jun Pang. An inductive approach to strand spaces. Formal Aspects of Computing, Vol. 25, No. 4, 2013.

[12CM] Cas Cremers, Sjouke Mauw. Operational Semantics and Verification of Security Protocols, Springer-Verlag Berlin Heidelberg, 2012.

[12G] Joshua D. Guttman. State and Progress in Strand Spaces: Proving Fair Exchange. Journal of Automated Reasoning, 48(2): 159-195, 2012.

[11CK] V. Cortier and S. Kremer, editors. Formal Models and Techniques for Analyzing Security Protocols, volume 5 of Cryptology and Information Security Series. IOS Press, 2011.

[11DMRS] A. Datta, J.C. Mitchell, A. Roy, S. Stiller, Protocol composition logic, in Formal Models and Techniques for Analyzing Security Protocols, ed. by V. Cortier, S. Kremer (IOS Press, Lansdale, 2011)

[11RS] Mark D. Ryan and Ben Smyth, Applied pi calculus, in: Formal Models and Techniques for Analyzing Security Protocols, Edited by Veronique Cortier, 2011 IOS Press, p. 112-142.

[08C] C.J.F. Cremers, On the protocol composition logic PCL, in ACM Symposium on Information, Computer & Communication Security (ASIACCS'08), ed. by M. Abe, V. Gligor, Tokyo, Japan (ACM, New York, 2008), pp. 66–76

[08CJSTW] Cervesato I., Jaggard A.D., Scedrov A., Tsay J.-K., Walstad C., Breaking and fixing public-key Kerberos, Information and Computation Volume 206, Issues 2-4, (2008), Pages 402-424.

[07ABF] M. Abadi, B. Blanchet, C. Fournet, Just Fast Keying in the Pi Calculus. In ACM Transactions on Information and System Security, 10(3), 2007.

[07DDMR] A. Datta, A. Derek, J.C. Mitchell, A. Roy, Protocol Composition Logic (PCL), in Computation, Meaning, and Logic: Articles dedicated to Gordon Plotkin, ed. by L. Cardelli, M. Fiore, G. Winskel. Electronic Notes in Theoretical Computer Science, vol. 172, (2007), pp. 311– 358

[07DGT1] S. Doghmi, J.D. Guttman, F.J. Thayer, Skeletons and the shapes of bundles, in 7th International Workshop on Issues in the Theory of Security (WITS'07), Braga, Portugal (2007)

[07DGT2] S.F. Doghmi, J.D. Guttman, F.J. Thayer, Skeletons, homomorphisms, and shapes: characterizing protocol executions, in 23rd Conference on the Mathematical Foundations of Programming Semantics (MFPS XXIII), New Orleans, USA. Electronic Notes in Theoretical Computer Science, vol. 173 (Elsevier, Amsterdam, 2007), pp. 85–102

[07DGT3] S.F. Doghmi, J.D. Guttman, F.J. Thayer, Searching for shapes in cryptographic protocols, in 13th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'07), ed. by O. Grumberg, M. Huth, Braga, Portugal. Lecture Notes in Computer Science, vol. 4424 (Springer, Berlin, 2007), pp. 523–537

[05AB] M. Abadi and B. Blanchet. Analyzing Security Protocols with Secrecy Types and Logic Programs. In Journal of the ACM, 52(1), pp. 102-146, 2005.

[05CDLMS] I. Cervesato, N. Durgin, P. Lincoln, J. Mitchell, A. Scedrov. A Comparison between Strand Spaces and Multiset Rewriting for Security Protocol Analysis. Journal of Computer Security, vol. 13, no. 2, pp. 265-316, 2005

[05KR] S. Kremer, M. Ryan. Analysis of an Electronic Voting Protocol in the Applied Pi Calculus. In 14th European Symposium on Programming (ESOP), pp. 186-200, 2005.

[02GT] J.D. Guttman, F.J. Thayer, Authentication tests and the structure of bundles. Theor. Comput. Sci. 283(2), 333–380 (2002)

[02SW] S.G. Stubblebine, R.N. Wright, An authentication logic with formal semantics supporting synchronization, revocation, and recency. IEEE Trans. Softw. Eng. 28(3), 256–285 (2002)

[01AF] M. Abadi, C. Fournet, Mobile values, new names, and secure communication, in 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'01), ed. by C. Hankin, D. Schmidt, London, UK (ACM, New York, 2001), pp. 104–115

[01B] B. Blanchet. An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In 14th IEEE Computer Security Foundations Workshop (CSFW), pp. 82-96, 2001.

[01DMP] N.A. Durgin, J.C. Mitchell, D. Pavlovic, A compositional logic for protocol correctness, in 14th IEEE Computer Security Foundations Workshop (CSFW'01), Cape Breton, Canada (IEEE Computer Society, Los Alamitos, 2001), pp. 241–272

[00AR] M. Abadi, P. Rogaway, Reconciling two views of cryptography (the computational soundness of formal encryption), in IFIP International Conference on Theoretical Computer Science (IFIP TCS'00), ed. by J. van Leeuwen, O. Watanabe, M. Hagiya, P.D. Mosses, T. Ito, Sendai, Japan (2000), pp. 3–22

[00B] G. Bella. Inductive Verification of Cryptographic Protocols. PhD thesis, Cambridge University, 2000.

[00GT2] J. D. Guttman and F. J. Thayer. Authentication tests and the normal, efficient penetrator. IEEE Computer Society Symposium on Research in Security and Privacy, 2000.

[00RSGLR] P.Y.A. Ryan, S.A. Schneider, M.H. Goldsmith, G. Lowe and A.W. Roscoe. The Modelling and Analysis of Security Protocols: the CSP Approach, Addison-Wesley, 2000.

[00RS] P. Y. A. Ryan and S. A. Schneider. Process algebra and non-interference. Journal of Computer Security, 2000.

[99AG] M. Abadi, A.D. Gordon, A calculus for cryptographic protocols: the Spi calculus. Inf. Comput. 148, 1–70 (1999)

[99P] L. C. Paulson. Inductive Analysis of the Internet Protocol TLS. In ACM Trans. on Information and System Security, 2(3), pp. 332-351, 1999.

[99THG1] F. J. Thayer, J. C. Herzog, and J. D. Guttman. Strand spaces: Proving security protocols correct. Journal of Computer Security, 7(2/3):191-230, 1999.

[99THG2] F.J. Thayer, J.C. Herzog, J.D. Guttman, Mixed Strand Spaces, in 12th IEEE Computer Security Foundations Workshop (CSFW'99), IEEE Computer Society, Los Alamitos, 1999, pp. 72–82

[98P] L.C. Paulson, The inductive approach to verifying cryptographic protocols. J. Comput. Secur. 6(1–2), 85–128 (1998)

[98THG1] F.J. Thayer, J.C. Herzog, J.D. Guttman, Honest ideals on Strand Spaces, in 11th IEEE Computer Security Foundations Workshop (CSFW'98), Rockport, USA (IEEE Computer Society, Los Alamitos, 1998), pp. 66–77

[98THG2] F. J. Thayer, J. C. Herzog, and J. D. Guttman. Strand spaces: why is a security protocol correct? IEEE Computer Society Symposium on Security and Privacy, 1998.

[98S] S. A. Schneider. Verifying authentication protocols in CSP. IEEE Transactions on Software Engineering, 1998.

[97DS] B. Dutertre and S. A. Schneider. Embedding CSP in PVS. An application to authentication protocols. Theorem proving in Higher Order Logics, number 1275 in LNCS. Springer, 1997.

[97LR] G. Lowe and A. W. Roscoe. Using CSP to detect errors in the TMN protocol. IEEE Transactions in Software Engineering, 23(10), 1997.

[97P] L.C. Paulson, Proving properties of security protocols by induction, in 10th IEEE Computer Security Foundations Workshop (CSFW'97), Rockport, Massachusetts (IEEE Computer Society, Los Alamitos, 1997), pp. 70–83

[96S] S. Schneider, Security properties and CSP, in 17th IEEE Symposium on Security & Privacy (S&P'96), Oakland, USA (IEEE Computer Society, Los Alamitos, 1996), pp. 174–187.

[96SvO] P.F. Syverson, P.C. van Oorschot, A unified cryptographic protocol logic. CHACS Report 5540-227 NRL (1996)

[96SS] S. A. Schneider and A. Sidiropoulos. CSP and anonymity. European Symposium on Research in Computer Security, 1996.

[95AN] R. Anderson and R. Needham. Programming Satan's computer. In J. van Leeuwen (ed.) Computer Science Today, volume 1000 of LNCS. Springer, 1995.

[95L] Gavin Lowe. An attack on the Needham-Schroeder public key authentication protocol. Information Processing Letters, 56(3):131–136, November 1995.

[94KMM] R.A. Kemmerer, C. Meadows, J.K. Millen, Three systems for cryptographic protocol analysis. J. Cryptol. 7, 79–130 (1994)

[93vO] P.C. van Oorschot, Extending cryptographic logics of belief to key agreement protocols, in 1st ACM Conference on Computer and Communications Security (ACM CCS'93), ed. by D.E. Denning, R. Pyle, R. Ganesan, R.S. Sandhu, V. Ashby, Fairfax, USA (ACM, New York, 1993), pp. 232–243

[93SM] Syverson P., Meadows C., A Logical Language for Specifying Cryptographic Protocol Requirements, Proceedings of the 1993 IEEE Computer Security Symposium on Security and Privacy, (1993) 165-177, IEEE Computer Society Press.

[91AT] M. Abadi, M. Tuttle, A semantics for a logic of authentication, in 10th ACM Symposium on Principles of Distributed Computing (PODC'91), Montreal, Canada (ACM, New York, 1991), pp. 201–216

[90BAN] Burrows M., Abadi M., Needham R., A Logic of Authentication. In ACM Transactions on Computer Systems, 8(1), (1990) 18-36.

[90GNY] L. Gong, R.M. Needham, R. Yahalom, Reasoning about belief in cryptographic protocol analysis, in 11th IEEE Symposium on Security & Privacy (S&P'90), Oakland, USA (IEEE Computer Society, Los Alamitos, 1990), pp. 234–248

[87NS] Needham R., Schroeder M., Authentication revisited, Operating Systems Review, Vol. 21, No. 1, (1987).

[85H] C. A. R. Hoare. Communicating Sequential Processes. Prentice-Hall, 1985.

[81DS] Denning D., Sacco G., Timestamps in Key Distribution Protocols, Communications of the ACM, Vol. 24, No. 8, (1981) 533-536.

[80M] R. Milner, A Calculus of Communicating Systems, Springer Verlag, 1980.

[78NS] Roger Needham and Michael Schroeder. Using encryption for authentification in large networks of computers. Communications of the ACM, 21(12), December 1978.

Mathematical model and methods of verification of cryptographic protocols

Mironov A.M.

In this paper, a new mathematical model of cryptographic protocols is presented, and examples of the application of this model for solving problems of verification of cryptographic protocols are given. Cryptographic protocols are distributed algorithms designed to enable the transmission of confidential information in an insecure environment. They are used, for example, in electronic payments, electronic voting procedures, systems for accessing confidential data, etc. Errors in cryptographic protocols can lead to great damage, therefore it is necessary to use mathematical methods to substantiate the various properties of correctness and security of cryptographic protocols. The paper outlines new methods for formal verification of cryptographic protocols. *Keywords:* cryptographic protocols, sequential processes, distributed processes, verification.