УДК 519.681.2

Процессный подход к верификации криптографических протоколов*

Миронов А.М. (Московский Государственный Университет имени М.В. Ломоносова)

В настоящей работе излагается новая математическая модель криптографических протоколов, и приводятся примеры применения этой модели для решения задач верификации криптографических протоколов. Криптографические протоколы – это коммуникационные протоколы, реализованные с применением криптографических алгоритмов для решения задач защиты информации, в рамках которого стороны информационного взаимодействия последовательно выполняют определенные действия и обмениваются сообщениями. 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. Для моделирования криптографических протоколов в работе вводятся понятия последовательного и распределенного процессов. Предлагаемый подход предназначен только для доказательства корректности криптографических протоколов. Особенностью модели протоколов является её простота по сравнению с другими моделями протоколов, основанных на логических формулах или на алгебраических процессных выражениях. Участники протоколов представляются в виде графов, представляющих системы переходов. Действия, выполняемые участниками, являются метками этих переходов. Методы обоснования корректности протоколов, рассматриваемые в настоящей статье, связаны с рассуждениями для графов, которые более просты и наглядны по сравнению с методами, основанными на построении логического вывода в логических и алгебраических моделях протоколов.

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

*при поддержке Математического центра в Академгородке, соглашение с Министерством науки и высшего образования Российской Федерации № 075-15-2025-349

1. Введение

Представленное исследование является развитием исследования, описанного в [2].

1.1. Понятие криптографического протокола

Криптографический протокол (КП) – это коммуникационный протокол, реализованный с применением криптографических алгоритмов для решения задач защиты информации, в рамках которого стороны информационного взаимодействия последовательно выполняют определенные действия и обмениваются сообщениями. 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 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 [122], 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 [111], связанная с возможностью непредусмотренного нечестного поведения одного из участников этого КП и нарушающая свойство безопасности 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 [70]): 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 [120], [118], [110], [78]), 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 конфиденциальность передаваемых сообщений (т.е. обоснование следующего свойства анализируемого КП: содержание сообщений, посланных одним участником этого КП другому участнику этого КП, 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 [74] и [72]. 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 [116] 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 [117], [115], [113], [114], [112], [108], [88]. Важным классом логических исчислений для моделирования и анализа КП является композиционная логика протоколов (Protocol Composition Logic), которой посвящены работы [91], [80], [77], [75]. 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 [90], [84] и др.

   Важным классом логических методов моделирования и анализа КП является индуктивный метод Паульсона: [106], [100], [97], [92].

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

   Источником данного класса моделей является основополагающая работа Р.Милнера [121]. 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 [96]. Среди других работ, относящихся к этому направлению, можно отметить работы [95], [89], [86], [79], [76], [67], [68], [66], [64].

3. Модели, основанные на **CSP**.

   CSP (**Communicating Sequential Processes**) – это математический аппарат, разработанный Ч.Хоаром [119] и предназначенный для моделирования и анализа распределенных вычислительных процессов. На базе этого аппарата построен метод моделирования и верификации КП, наиболее полно изложенный в книге [94]. Дедуктивная верификация КП на основе данного подхода использует понятие **ранг-функции**. Среди работ, относящихся к данному направлению, можно отметить работы [109], [107], [105], [104], [103], [65].

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 моделирования и верификации КП на основе понятия пространства нитей, можно отметить работы [101], [102], [98], [99], [93], [87], [85], [81], [82], [83], [73], [71], [69].

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.

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.

Основное достоинство предложенной модели заключаются в том, что доказательства свойств корректности КП на основе данной модели м.б. существенно короче, чем доказательства этих свойств на основе других моделей КП. Для обоснования этого утверждения мы приводим пример верификации КП Yahalom [94]. Верификация этого КП в вышеупомянутом источнике занимает несколько десятков страниц, в то время как верификация КП Yahalom на базе предложенной модели занимает менее 4 страниц. Причина такого существенного упрощения верификации КП связана с использованием доказанных в настоящей работе теорем 1 и 2, которые представляют собой схемы обоснования свойства защищённости сообщений при выполнении переходов протокола, и устраняют дублирование рассуждений при доказательстве корректности КП.

Предложенный формализм предназначен только для доказательства корректности криптографических протоколов, и не предназначен для обнаружения ошибок в некорректных КП. Если протокол является некорректным, то для обнаружения ошибок в нем должен использоваться другой метод, аналогичный методу Model Checking в теории верификации программ, см. [125]. Данный метод будет изложен в последующих публикациях автора.

Изложенный в статье язык описания РП имеет самостоятельную ценность, и может рассматриваться как новый язык описания распределенных алгоритмов с применением криптографических функций.

В целях простоты изложения в работе рассматривается такая математическая модель КП, которая отражает простейшие криптографические примитивы, используемые в КП: в ней формализуются лишь симметричные системы шифрования, и не рассматриваются такие примитивы как системы шифрования с открытым ключом, хэш-функции, цифровые подписи, и т.п. Все эти примитивы несложно ввести в представленную модель путем соответствующих дополнений.

2. Вспомогательные понятия

В этом параграфе мы излагаем понятия, необходимые для определения понятий последовательного и распределённого процесса.

2.1. Типы, переменные, константы и функциональные символы

Будем предполагать, что заданы следующие множества.

- Множество $Types$, его элементы называются **типами данных** (или просто **типами**). Каждому типу τ из $Types$ сопоставлено множество D_τ **значений** типа τ.
- Множества Var и Con, их элементы называются **переменными** и **константами** соответственно. Каждой переменной $x \in Var$ и константе $c \in Con$ сопоставлен тип $\tau(x)$ и $\tau(c) \in Types$ соответственно. Каждая переменная x может принимать **значения** в некотором множестве, которое будем обозначать $D_{\tau(x)}$, т.е. в различные моменты времени переменная x может быть связана с различными элементами множества $D_{\tau(x)}$.
- Множество Fun, его элементы называются **функциональными символами (ФС)**. Каждому $f \in Fun$ сопоставлен **функциональный тип (ФТ)** $\tau(f)$, который представляет собой запись вида

$$(\tau_1, \ldots, \tau_n) \to \tau, \text{ где } \tau_1, \ldots, \tau_n, \tau \in Types. \tag{1}$$

Будем считать, что среди типов, входящих в указанное выше множество $Types$, присутствуют следующие типы:

- **A**, значения этого типа называются **агентами**,
- **K**, значения этого типа обозначают **ключи**, используемые агентами для шифрования или расшифрования сообщений,
- **M**, значения этого типа обозначают **сообщения**, которые агенты могут пересылать друг другу во время своей работы,
- для каждого типа τ множество $Types$ содержит тип $\mathbf{2}^\tau$, значениями которого являются подмножества множества D_τ, тип $\mathbf{2}^\tau$ используется, например, для представления содержимого канала: его значениями могут быть множества различных сообщений, содержащихся в канале,
- для каждого списка типов $\tau_1, \ldots, \tau_n$ множество $Types$ содержит тип $(\tau_1, \ldots, \tau_n)$, и $D_{(\tau_1,\ldots,\tau_n)} = D_{\tau_1} \times \ldots \times D_{\tau_n}$.

2.2. Термы

Термы строятся из переменных, констант и ФС. Множество всех термов обозначается символом Tm. Каждый терм e имеет тип $\tau(e) \in Types$, определяемый структурой терма e.

Правила построения термов имеют следующий вид:

- если $x \in Var \cup Con$, то x – терм типа $\tau(x)$, и
- если $e_1, \ldots, e_n \in Tm$, $f \in Fun$, и $\tau(f)$ имеет вид (1), где $\tau_i = \tau(e_i)$ $(i = 1, \ldots, n)$, то запись $f(e_1, \ldots, e_n)$ – терм типа τ.

Терм $e \in Tm$ называется **подтермом** терма $e' \in Tm$, если либо $e = e'$, либо $e' = f(e_1, \ldots, e_n)$, 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 терма e'. Запись $e \subset e'$, где $e, e' \in Tm$, означает, что $e \subseteq e'$ и $e \neq e'$.

Индукцией по структуре терма e нетрудно доказать, что

$$\begin{array}{l}\text{если } e_1 \text{ и } e_2 \text{ – различные подтермы терма } e,\\ \text{то либо } e_1 \subset e_2, \text{ либо } e_2 \subset e_1, \text{ либо } e_1 \text{ и } e_2 \text{ не имеют общих компонентов.}\end{array} \qquad (2)$$

Будем считать, что Fun содержит следующие ФС:

- ФС *encrypt* типа $(\mathbf{K}, \mathbf{M}) \to \mathbf{M}$,
  терм вида $encrypt(k, e)$ обозначает сообщение, получаемое шифрованием сообщения e на ключе k в симметричной системе шифрования, будем обозначать такой терм записью $k(e)$ и называть его **шифрованным сообщением (ШС)**,
- ФС *shared_key* типа $(\mathbf{2^A}) \to \mathbf{K}$,
  терм вида $shared\_key(\{A_1, \ldots, A_n\})$ называется **разделяемым ключом** агентов $A_1, \ldots, A_n$ и будет обозначаться $k_{A_1,\ldots,A_n}$,
- ФС *list* типа $(\tau_1, \ldots, \tau_n) \to (\tau_1, \ldots, \tau_n)$, где $(\tau_1, \ldots, \tau_n)$ – произвольные типы, терм вида $list(e_1, \ldots, e_n)$ обозначает список термов, компоненты которого – термы $e_1, \ldots, e_n$, мы будем обозначать терм $list(e_1, \ldots, e_n)$ более короткой записью $(e_1, \ldots, e_n)$.

Отметим, что мы не предполагаем наличие в Fun ФС *decrypt*, обозначающего операцию расшифрования в симметричной системе шифрования. Это связано с тем, что операцию расшифрования шифртекста e на ключе k можно выразить в нашем языке путем действия вида $k(x) := e$, которое заключается в нахождении такого значения переменной x, результат шифрования которого на ключе k будет совпадать с значением терма e (понятие действия изложено в пункте 3).

Поясним также, каким образом в рассматриваемой модели осуществляется проверка возможности доступа участника к зашифрованной информации. Согласно определению

понятия действия вида $e := e'$, операция расшифрования может быть выполнена участником в том и только том случае когда ключ, который он использует для расшифрования, совпадает с тем ключом, на котором было зашифровано расшифровываемое сообщение. Если эти ключи разные, то расшифрование не может быть выполнено, т.е. выполнение протокола не дойдёт до своего заключительного состояния.

В предложенном формализме функциональные символы являются абстракцией идеальных криптографических примитивов, поэтому для полной гарантии корректности криптографических протоколов необходимо гарантировать корректность реализаций таких криптографических примитивов. Для гарантии корректности реализаций таких криптографических примитивов применяются методы дедуктивной верификации программ, например, дедуктивная верификация применялась для проверки корректности реализаций хэш-функций SHA-256 и «Стрибог», что описано в статьях [11] и [1] соответственно.

Также отметим, что мы не рассматриваем формализацию в нашей модели таких компонентов криптографических протоколов как асимметричное шифрование, цифровая подпись, хэш-функции, схемы разделения секрета, и т. п. Формализация всех этих компонентов является несложной и производится на основе введения дополнительных ФС.

Будем использовать следующие обозначения:

- $\forall\, x \in Var, \forall\, e \in Tm$ запись $x \in e$ означает, что x входит в e,
- $\forall\, E \subseteq Tm\ \ Var(E) = \{x \in Var \mid \exists\, e \in E : x \in e\}$,
- $\forall\, E \subseteq Tm$ запись $Tm(E)$ обозначает наименьшее по включению множество, удовлетворяющее следующим условиям:
  - $E \subseteq Tm(E)$, $Con \subseteq Tm(E)$,
  - для каждого терма вида $f(e_1, \ldots, e_n)$

    если $e_1, \ldots, e_n \in Tm(E)$, то $f(e_1, \ldots, e_n) \in Tm(E)$,
- $\forall\, \tau \in Types$, $\forall\, E \subseteq Tm\ E_\tau = \{e \in E \mid \tau(e) = \tau\}$.

2.3. Формулы и теории

Элементарной формулой (ЭФ) называется запись одного из следующих видов:

$$e = e', \quad \text{где } \tau(e) = \tau(e'),$$
$$e \in e', \quad \text{где } \tau(e') = \mathbf{2}^{\tau(e)}.$$

Формулой называется обычная булева комбинация ЭФ, в которой могут использоваться логические связки $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$ и константы 1 и 0. Множество всех формул обознача-

ется Fm. Формулы вида $\wedge(e_1, e_2)$ будем записывать в более привычном виде $e_1 \wedge e_2$, аналогичная запись используется для других булевых комбинаций. Формулы вида $e_1 \wedge \ldots \wedge e_n$ могут также записываться в виде $\left\{ \begin{array}{l} e_1 \\ \ldots \\ e_n \end{array} \right\}$ или $\{e_1, \ldots, e_n\}$. Возможна конъюнкция произвольного семейства формул $\{e_i \mid i \in I\}$, она обозначается записью $\bigwedge_{i \in I} e_i$.

Теорией называется произвольная совокупность формул $Th \subseteq Fm$. Понятие **доказуемости** формулы в теории определяется стандартным образом. Ниже мы будем предполагать, что задана некоторая теория Th, и будем говорить что формула φ **доказуема**, если она доказуема в этой теории Th.

2.4. Подстановки

Подстановкой называется функция $\theta : Var \to Tm$, такая, что

$$\forall\, x \in Var \;\; \tau(x) = \tau(\theta(x)).$$

Будем говорить, что θ заменяет переменную $x \in Var$ на терм $\theta(x)$. Подстановка называется **тождественной**, если $\forall\, x \in Var \;\; \theta(x) = x$.

Будем использовать следующие обозначения:

- Θ обозначает множество всех подстановок,
- $\forall\, \theta \in \Theta,\ \forall\, e \in Tm$ запись e^θ обозначает терм, получаемый из e заменой для каждой переменной $x \in Var(e)$ каждого вхождения x в e на терм $\theta(x)$;
- $\forall\, \theta \in \Theta,\ \forall\, E \subseteq Tm\ E^\theta = \{e^\theta \mid e \in E\}$.
- $\forall\, \theta \in \Theta, \forall\, \varphi \in Fm$ запись $\theta \vdash \varphi$ обозначает утверждение, что формула φ^θ доказуема.

3. Последовательные и распределенные процессы

В этой главе определяются понятия последовательного и распределенного процесса. Последовательный процесс является моделью участника КП, а распределенный процесс является моделью всего КП. Предложенная модель является теоретической основой для метода верификации КП, излагаемого в пункте 6.

3.1. Действия

Действие – это запись одного из следующих видов:

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

которые называются **выводом** сообщения e в открытый канал $\circ$, **вводом** сообщения e из открытого канала $\circ$, и **присваиванием**, соответственно. Множество всех действий обозначается Act.

3.2. Последовательные процессы

Последовательным процессом (или просто **процессом**) будем называть граф P со следующими свойствами:

- P имеет выделенные вершины $\odot$ и $\otimes$, называемые **начальной** и **терминальной** вершинами соответственно, из $\otimes$ не выходят рёбра,
- каждому ребру графа P сопоставлена метка $a \in Act$, ребро процесса P представляется записью $v \xrightarrow{a} v'$, где v и v' – начало и конец ребра, a – метка ребра.

Процесс является описанием поведения дискретной динамической системы, работа которой заключается в последовательном выполнении действий, связанных с вводом и выводом сообщений и изменением значений переменных. С каждым процессом P связаны

- **агент** $Agent_P \in Var_{\mathbf{A}}$, называемый **исполнителем** процесса P,
  напомним, что согласно обозначениям, введенным в конце пункта 2.2, $Var_{\mathbf{A}}$ – это множество переменных, имеющих тип **A** (агент),
- множество Var_P **переменных** процесса P, являющееся дизъюнктным объединением следующих множеств:
  - множество $Public_P$ **открытых переменных**,
  - множество $Private_P$ **приватных переменных**,
  - множество $Unique_P$ переменных, инициализированных уникальными значениями, эти переменные обозначают криптографические ключи, или переменные, называемые **нонсами**,
  - $\{x_P\}$, значения x_P – подмножества множества

$$Public_P \cup Private_P \cup Unique_P,$$

    их элементы называются **инициализированными переменными** процесса P,
  - $\{at_P\}$, значения at_P – вершины графа P,
  - $\{x_\circ\}$, $\tau(x_\circ) = \mathbf{2^M}$, значения $x_\circ$ интерпретируются как **содержимое открытого канала** (отметим, что переменная $x_\circ$ является общей для всех процессов).

В каждый момент выполнения процесса каждая его неслужебная переменная (т.е. отличная от at_P, x_P, $x_\circ$) либо инициализирована каким-либо значением, либо неинициализи-

рована. Если она в какой-либо момент времени инициализирована, то во все последующие моменты данная переменная имеет то же значение что и в момент инициализации. Переменные из $Unique_P$ инициализированы в начальный момент.

3.3. Процесс противника

Процесс противника – это процесс †, обладающий свойствами:

- множество вершин графа процесса † одноэлементно,
- $\forall\, a \in Act$ граф процесса † содержит ребро с меткой a.

Ниже будем предполагать, что † – единственный из всех рассматриваемых процессов, граф которого имеет циклы.

3.4. Распределенные процессы

Распределенным процессом (РП) называется семейство процессов $\mathcal{P} = \{P_i \mid i \in I\}$, таких, что компоненты семейства

$$\{Private_{P_i} \cup Unique_{P_i} \cup \{x_{P_i}, at_{P_i}\} \mid i \in I\} \tag{3}$$

дизъюнктны (если это условие не выполняется, то соответствующие переменные в процессах P_i переименовываются).

С каждым РП $\mathcal{P}$ связано **начальное состояние** $\theta^0_{\mathcal{P}} \in \Theta$, обладающее следующими свойствами:

$$\forall\, P \in \mathcal{P} \quad x_P^{\theta^0_{\mathcal{P}}} = Public_P \cup Unique_P,\ at_P^{\theta^0_{\mathcal{P}}} = \odot, x_\circ^{\theta^0_{\mathcal{P}}} = \emptyset.$$

Если РП состоит из одного процесса P, то он обозначается тем же символом P. Если $\{\mathcal{P}_i \mid i \in I\}$ – семейство РП, то данная запись обозначает также РП, состоящий из всех процессов, входящих в какой-либо РП из семейства $\mathcal{P}_i$ ($\forall\, i \in I$).

Записи $Public_{\mathcal{P}}$ и $Unique_{\mathcal{P}}$ обозначают соответственно множества

$$\bigcup_{P\in\mathcal{P}} Public_P \quad \text{и} \quad \bigcup_{P\in\mathcal{P}} Unique_P.$$

3.5. Переходы в распределенных процессах

Переход в РП $\mathcal{P}$ – это утверждение, обозначаемое записью $\theta \xrightarrow{a_P} \theta'$, где $P \in \mathcal{P}$, $\theta, \theta' \in \Theta$ (θ называется **началом** данного перехода, а θ' – его **концом**) и a – метка некоторого ребра $v \xrightarrow{a} v'$ процесса P, причём выполнены условия:

1. $at_P^\theta = v$, $at_P^{\theta'} = v'$,
2. $\forall\, x \in x_P^\theta \setminus \{at_P, x_P, x_\circ\}\ \ x^\theta = x^{\theta'}$,
3. если $a = \circ!e$, то
   - $e \in Tm(x_P^\theta), x_P^{\theta'} = x_P^\theta, x_\circ^{\theta'} = x_\circ^\theta \cup \{e^\theta\}$,
   - если e^θ содержит подтерм вида $k(\tilde{e})$, где $k \in Tm_{\mathbf{K}}$, то верно следующее свойство:

$$\text{либо } k \in Var, \text{ либо } \left\{ \begin{array}{l} k = shared\_key(\ldots) \\ Agent_P \in k \end{array} \right\}, \tag{4}$$

   напомним, что согласно обозначениям, введенным в конце пункта 2.2, $Tm_{\mathbf{K}}$ – это множество термов, имеющих тип **K** (ключ), и $Agent_P \in k$ означает, что переменная $Agent_P$ входит в терм k,
4. если $a = \circ?e$ или $e := e'$, то
   (a) $x_\circ^{\theta'} = x_\circ^\theta$, $x_P^{\theta'} = x_P^\theta \cup Var(e)$,
   (b) $\theta' \vdash e \in x_\circ$ или $\left\{ \begin{array}{l} \theta' \vdash e = e' \\ e' \in Tm(x_P^\theta) \end{array} \right\}$ соответственно,
   (c) если e^θ содержит подтерм вида $k(\tilde{e})$, где $k \in Tm_{\mathbf{K}}$, то верно свойство (4),
   (d) если $a = (e := e')$ и $k = shared\_key(\ldots)$, то верна импликация

$$k \subseteq (e')^\theta \ \Rightarrow\ Agent_P \in k. \tag{5}$$

Переход $\theta \xrightarrow{a_P} \theta'$ РП $\mathcal{P}$ интерпретируется как выполнение процессом $P \in \mathcal{P}$ действия a, в результате чего $\mathcal{P}$ переходит от θ к θ'. Если в текущий момент с $\mathcal{P}$ связана подстановка θ, и в этот момент некоторый процесс P, входящий в $\mathcal{P}$, содержит ребро $v \xrightarrow{a} v'$, причем $v = at_P^\theta$, то мы считаем, что РП $\mathcal{P}$, связанный в текущий момент с подстановкой θ, может выполнить действие a, после чего он будет связан с подстановкой θ', удовлетворяющей вышеприведённым условиям при этом

- при выполнении действия $\circ!e$ происходит добавление терма e^θ к содержимому открытого канала $\circ$,
- при выполнении действия $\circ?e$ или $e := e'$ происходит либо чтение некоторого терма из содержимого канала $\circ$, либо присваивание соответственно, путем инициализации неинициализированных в текущий момент переменных из терма e: терм e рассматривается как шаблон, которому должен соответствовать некоторый терм из $x_\circ^\theta$ или терм $(e')^\theta$ соответственно, и выполняемое действие заключается в преобразовании θ в подстановку θ' путем определения подходящих значений переменных из $Var(e) \setminus x_P^\theta$,

с таким расчётом, чтобы значение $e^{\theta'}$ было бы равно некоторому терму из $x_\circ^\theta$ или терму $(e')^\theta$ соответственно.

3.6. Выполнение распределенного процесса

Выполнение РП $\mathcal{P}$ – это последовательность подстановок $\pi = (\theta_0, \theta_1, \ldots)$ РП $\mathcal{P}$, такая, что θ_0 – начальное состояние РП $\mathcal{P}$, и для каждой пары θ_i, θ_{i+1} соседних членов этой последовательности имеется переход $\theta_i \overset{a_P}{\to} \theta_{i+1}$, где P – какой-либо процесс из $\mathcal{P}$.

Для каждого выполнения $\pi = (\theta_0, \theta_1, \ldots)$ запись $\theta \in \pi$ означает, что $\exists\, i \geq 0 : \theta_i = \theta$.

Если задано выполнение $\pi = (\theta_0, \theta_1, \ldots)$ и θ, θ' – подстановки, входящие в π, то запись $\theta <_\pi \theta'$ означает, что $\theta = \theta_i$ и $\theta' = \theta_j$ для некоторых индексов $i < j$. Запись $\theta \leq_\pi \theta'$ означает, что $\theta <_\pi \theta'$ или $\theta = \theta'$.

Подстановка θ РП $\mathcal{P}$ называется **достижимым состоянием** РП $\mathcal{P}$, если она входит в некоторое выполнение $\mathcal{P}$. Множество всех достижимых состояний РП $\mathcal{P}$ обозначается $\Theta_\mathcal{P}$.

В начальный момент выполнения РП $\mathcal{P}$ переменные из $Unique_\mathcal{P}$ инициализированы **уникальными значениями**, т.е. такими значениями, которые никогда не встречались среди всех значений, используемых до начала выполнения $\mathcal{P}$.

4. Свойство защищённости

4.1. Определение свойства защищённости

В рассуждениях, связанных с верификацией РП, будем использовать свойство **защищённости**, обозначаемое записью

$$E \perp P, \text{ где } \begin{cases} E \subseteq Public_\mathcal{P} \cup Tm(Public_\mathcal{P})_\mathbf{K}, \\ P \in \mathcal{P}, \forall\, k \in E_\mathbf{K} \;\; Agent_P \notin k, \end{cases} \tag{6}$$

где $\mathcal{P}$ – некоторый РП. Напомним, что согласно обозначениям, введенным в конце пункта 2.2, $E_\mathbf{K}$ – это множество термов из E типа $\mathbf{K}$ (ключ).

Свойство (6) истинно в состоянии $\theta \in \Theta_\mathcal{P}$ (что обозначается записью $\theta \models E \perp P$), если

$$\begin{array}{l} \forall\, e \in E,\ \forall\, e' \in (x_P^\theta)^\theta \cup x_\circ^\theta \text{ каждое вхождение } e \text{ в } e' \\ \text{содержится в подтерме } k(\ldots) \subseteq e',\ \text{где } k \in E_\mathbf{K}. \end{array} \tag{7}$$

Данное свойство имеет следующий смысл: термы из E доступны процессу P в состоянии θ только в «защищённом» виде, т.е. содержатся в термах из $(x_P^\theta)^\theta \cup x_\circ^\theta$ только внутри ШС, которые зашифрованы на ключах, недоступных для P в θ.

4.2. Сохранение защищённости при переходах

В этом параграфе доказывается теорема о сохранении свойства защищённости $E \perp P$ при переходах РП. Данная теорема м.б. интерпретирована как следующее утверждение: если в текущем состоянии θ верно свойство $E \perp P$, то никакая собственная активность процесса P, начиная с состояния θ, не приведет к тому, что какое-либо сообщение из E когда-нибудь станет доступным процессу P.

Теорема 1

Пусть задан переход $\theta \xrightarrow{a_P} \theta'$ в РП $\mathcal{P}$.

$\forall\, E \subseteq Public_{\mathcal{P}} \cup Tm(Public_{\mathcal{P}})_{\mathbf{K}}$ верна импликация

$$\theta \models E \perp P \Rightarrow \theta' \models E \perp P. \tag{8}$$

Доказательство.

Пусть верна посылка импликации (8), т.е. верно (7). Докажем, что тогда будет верно её заключение, т.е.

$$\begin{array}{l} \forall\, e \in E,\ \forall\, e' \in (x_P^{\theta'})^{\theta'} \cup x_\circ^{\theta'} \text{ каждое вхождение } e \text{ в } e' \\ \text{содержится в подтерме } k(\tilde{u}) \subseteq e',\ \text{где } k \in E_{\mathbf{K}}. \end{array} \tag{9}$$

Если (9) неверно, то

$$\begin{array}{l} \exists\, e \in E, \exists\, e' \in (x_P^{\theta'})^{\theta'} \cup x_\circ^{\theta'}, \exists \text{ вхождение } e \subseteq e',\ \text{которое} \\ \text{не содержится в каждом терме вида } k(\tilde{u}) \subseteq e', \text{где } k \in E_{\mathbf{K}}. \end{array} \tag{10}$$

Рассмотрим три варианта перехода $\theta \xrightarrow{a_P} \theta'$.

1. $a_P = \circ!u$, $u \in Tm(x_P^\theta), x_P^{\theta'} = x_P^\theta, x_\circ^{\theta'} = x_\circ^\theta \cup \{u^\theta\}$.

   В этом случае (10) возможно только если $e' = u^\theta$.

   Возможен один из следующих двух случаев:

   (a) $u \in x_P^\theta$, в этом случае из $e \subseteq u^\theta = e'$ и (7) следует

   $$e \subseteq k(\tilde{u}) \subseteq u^\theta = e',\ \text{ где } k \in E_{\mathbf{K}},$$

   что противоречит (10),

   (b) $u = f(u_1, \ldots, u_n)$, где $f \in Fun$, в этом случае противоречивость (10) обосновывается индукцией по структуре u: в случае $e \subset u^\theta$ противоречивость (10) следует

из индуктивного предположения, а в случае $e = u^\theta = e'$, согласно свойству (7), e содержится в подтерме $k(\tilde{u}) \subseteq e' = e$, т.е. $e = k(\tilde{u})$, что невозможно по условию из (6) на множество E.

2. $a_P = \circ?u$, $x_\circ^{\theta'} = x_\circ^\theta$, $x_P^{\theta'} = x_P^\theta \cup Var(u)$, $u^{\theta'} \in x_\circ^\theta$, и если u^θ содержит подтерм $k(\tilde{u})$, где $k \in Tm_{\mathbf{K}}$, то верно (4).

   В этом случае (10) возможно только если

$$\begin{array}{l} \exists\, e \in E,\ \exists\, y \in Var(u) \setminus x_P^\theta : e \subseteq y^{\theta'} = e', \\ \text{и не существует терма вида } k(\tilde{u}), \text{ где } k \in E_{\mathbf{K}}, \\ \text{такого, что } e \subseteq k(\tilde{u}) \subseteq y^{\theta'}. \end{array} \tag{11}$$

   Поскольку упомянутый в (11) терм e содержится в терме $y^{\theta'} \subseteq u^{\theta'} \in x_\circ^\theta$, то из (7) следует, что e содержится в некотором подтерме вида $k(\tilde{u}) \subseteq u^{\theta'}$, где $k \in E_{\mathbf{K}}$.

   Таким образом, $e \subseteq k(\tilde{u})$ и $e \subseteq y^{\theta'}$, т.е. термы $k(\tilde{u})$ и $y^{\theta'}$ имеют непустое пересечение, поэтому, согласно (2), либо $k(\tilde{u}) \subseteq y^{\theta'}$, либо $y^{\theta'} \subset k(\tilde{u})$. Включение $k(\tilde{u}) \subseteq y^{\theta'}$ противоречит (11), следовательно $y^{\theta'} \subset k(\tilde{u})$, поэтому

$$y^{\theta'} \subset k(\tilde{u}) \subseteq u^{\theta'}. \tag{12}$$

   Докажем индукцией по структуре терма u, что из правого включения в (12) следует, что

$$\exists\, z \in Var(u) : k(\tilde{u}) \subseteq z^{\theta'} \subseteq u^{\theta'}. \tag{13}$$

   Если $u \in Var$, то $z = u$, если $u \in Con$, то (12) неверно.

   Пусть $u = f(u_1, \ldots, u_n)$, где $f \in Fun$, тогда возможны следующие случаи:

   - $f = encrypt$, т.е. $u = k_1(u_1)$: если $k_1 \in Var_{\mathbf{K}}$, то возможны следующие случаи:
     - $k(\tilde{u}) = u^{\theta'} = k_1^\theta(u_1^{\theta'})$, в этом случае $k = k_1^\theta \in E$, и, согласно (4), либо $k \in Var$, либо

$$k = shared\_key(\ldots) \text{ и } Agent_P \in k. \tag{14}$$

       Случай $k \in Var$ невозможен, т.к. из свойств $k \in (x_P^\theta)^\theta$ и $k \in E_{\mathbf{K}}$, согласно (7), следует, что вхождение k в k должно содержаться в подтерме вида $k'(\ldots) \subseteq k$, что невозможно, а (14) невозможно согласно второй строке в (6),
     - $k(\tilde{u}) \subseteq k_1^{\theta'}$, данный случай невозможен по определению термов типа $\mathbf{K}$,
     - $k(\tilde{u}) \subseteq u_1^{\theta'}$, в данном случае утверждение (13) следует из индуктивного предположения,

а если $k_1 = shared\_key(\ldots)$, то верно (14), что невозможно согласно второй строке в (6),

- если $f = list$, то $\exists\, i \in 1, \ldots, n : k(\tilde{u}) \subseteq u_i^{\theta'}$, и (13) следует из индуктивного предположения,
- случай $f = shared\_key$ невозможен.

Из (12) и (13) следует, что

$$y^{\theta'} \subset k(\tilde{u}) \subseteq z^{\theta'} \subseteq u^{\theta'}. \tag{15}$$

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

3. $a_P = (u := u')$, $x_\circ^{\theta'} = x_\circ^\theta$, $x_P^{\theta'} = x_P^\theta \cup Var(u)$, $u' \in Tm(x_P^\theta)$, $u^{\theta'} = (u')^\theta$, и если u^θ содержит подтерм вида $k(\tilde{u})$, где $k \in Tm_{\mathbf{K}}$, то верно (4).

   Анализ данного случая аналогичен анализу предыдущего случая. Свойство (10) верно только если верно свойство (11), из которого следует, что терм e, упомянутый в (11), обладает свойством $e \subseteq u^{\theta'} = (u')^\theta$.

   Докажем, что

$$\exists\, v \in Var(u') : e \subseteq v^\theta. \tag{16}$$

   Т.к. $e \in E$, то, согласно (6),

   - либо $e \in Var$, в этом случае (16) очевидно,
   - либо e имеет вид $shared\_key(\ldots)$, в этом случае из $e \subseteq (u')^{\theta'}$, соотношения (5) и второй строки в (6) следует, что данный случай невозможен.

   Т.к. $u' \in Tm(x_P^\theta)$, т.е. $Var(u') \subseteq x_P^\theta$, то из (16) следует, что $v \in x_P^\theta$, и $e \subseteq v^\theta \in (x_P^\theta)^\theta$, откуда, на основании (7), заключаем, что $e \subseteq k(\tilde{u}) \subseteq v^\theta \subseteq (u')^\theta = u^{\theta'}$, где $k \in E_{\mathbf{K}}$.

   Термы $k(\tilde{u})$ и $y^{\theta'}$ имеют непустое пересечение (оба содержат e), и из (11) следует, что $k(\tilde{u})$ не м.б. подтермом терма $y^{\theta'}$, поэтому из (2) следует, что

$$y^{\theta'} \subset k(\tilde{u}) \subseteq u^{\theta'}. \tag{17}$$

   Так же, как и в предыдущем случае, доказываем, что из (17) следует свойство

$$\exists\, z \in Var(u) : k(\tilde{u}) \subseteq z^{\theta'} \subseteq u^{\theta'}. \tag{18}$$

   Из (17) и (18) следует, что

$$y^{\theta'} \subset k(\tilde{u}) \subseteq z^{\theta'} \subseteq u^{\theta'}. \tag{19}$$

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

5. Свойство соответствия

В этом параграфе формулируется и доказывается теорема, которая может использоваться для обоснования **свойства соответствия** протоколов аутентификации. Данное свойство имеет следующий неформальный смысл: если один из участников протокола аутентификации (обозначим его A) после выполнения этого протокола пришел к выводу, что другой участник этого протокола (обозначим его B) является подлинным (т.е. те параметры, которые получил A от якобы участника B, совпадают с теми параметрами, которые B посылал A), то B действительно посылал A сообщение с этими параметрами.

Доказываемая ниже теорема имеет следующий смысл: если при некотором выполнении π РП $\mathcal{P}$ в состоянии $\theta \in \pi$ в канале $\circ$ содержится сообщение, содержащее подтерм $k(e)$, где ключ k недоступен в состоянии θ для некоторого процесса $P \in \mathcal{P}$, то в некотором состоянии $\theta' <_\pi \theta$ другой процесс $P' \neq P$ из $\mathcal{P}$ послал в канал $\circ$ сообщение, содержащее $k(e)$.

Теорема 2

Пусть заданы

- РП $\mathcal{P}$ и некоторое его выполнение $\pi = (\theta_0, \theta_1, \ldots)$,
- подмножество $E \subseteq Public_{\mathcal{P}} \cup Tm(Public_{\mathcal{P}})_{\mathbf{K}}$, и
- состояние $\theta \in \pi$, причем $\theta \models E \bot P$, и $\exists\, e \in x^\theta_\circ$:

$$\exists\, k(\tilde{e}) \subseteq e, \text{ где } k \in E_{\mathbf{K}}. \tag{20}$$

Тогда $\exists\, P' \in \mathcal{P} : P' \neq P$ и π содержит переход вида

$$\dot{\theta} \overset{(\circ!\dot{e})_{P'}}{\longrightarrow} \theta', \text{ где } k(\tilde{e}) \subseteq \dot{e}^{\dot{\theta}} \text{ и } \theta' \leq_\pi \theta. \tag{21}$$

Доказательство.

Пусть θ' – первое состояние на π, такое, что $\exists\, e' \in x^{\theta'}_\circ$:

$$k(\tilde{e}) \subseteq e', \text{ где } k(\tilde{e}) \text{ – терм из (20).} \tag{22}$$

Нетрудно видеть, что $\theta' \leq_\pi \theta$, и т.к. $x^{\theta_0}_\circ = \emptyset$, то $\theta' \neq \theta_0$.

Пусть переход на пути π с концом в θ' имеет вид $\dot{\theta} \overset{a_{P'}}{\to} \theta'$. Т.к. $e' \notin x_\circ^{\dot{\theta}}$, то $a'_P = \circ!\dot{e}$, где $\dot{e}^{\dot{\theta}} = e'$. Если $P' \neq P$, то теорема доказана.

Докажем, что случай $P' = P$ невозможен.

Пусть $P' = P$, т.е. $\dot{\theta} \overset{(\circ!\dot{e})_P}{\longrightarrow} \theta'$.

Если $k = shared\_key(\ldots)$, то, согласно (22) и (4), $Agent_P \in k \in E_{\mathbf{K}}$, что противоречит второй строке в (6). Поэтому $k \in Var$.

Из $\theta \models E \bot P$ следует, что $\dot{\theta} \models E \bot P$.

Случай $k \in (x_P^{\dot{\theta}})^{\dot{\theta}}$ невозможен, т.к. из $k \in (x_P^{\dot{\theta}})^{\dot{\theta}}$ и $k \in E_{\mathbf{K}}$, согласно (7), следует, что вхождение k в k должно содержаться в подтерме вида $k'(\ldots) \subseteq k$, что невозможно.

Докажем индукцией по структуре терма $\dot{e}$, что из свойства

$$k(\tilde{e}) \subseteq \dot{e}^{\dot{\theta}}, k \in E_{\mathbf{K}}, k \in Var \text{ и } k \notin (x_P^{\dot{\theta}})^{\dot{\theta}} \tag{23}$$

следует утверждение

$$\exists\, x \in Var(\dot{e}) \subseteq x_P^{\dot{\theta}} : k(\tilde{e}) \subseteq x^{\dot{\theta}}. \tag{24}$$

Если $\dot{e} \in Var$, то $x = \dot{e}$, а если $\dot{e} \in Con$, то (23) неверно.

Пусть $\dot{e} = f(e_1, \ldots, e_n)$, где $f \in Fun$, тогда

- если $f = encrypt$, т.е. $\dot{e} = k_1(e_1)$, то возможны следующие случаи:
  - $k(\tilde{e}) = \dot{e}^{\dot{\theta}} = k_1^{\dot{\theta}}(e_1^{\dot{\theta}})$, в этом случае
    - $k = k_1^{\dot{\theta}}$, и если k_1 имеет вид $shared\_key(\ldots)$, то $k = k_1^{\dot{\theta}}$ тоже имеет такой вид, но по предположению (23) $k \in Var$, и
    - если $k_1 \in Var$, то $k_1 \in x_P^{\dot{\theta}}$, т.е. $k = k_1^{\dot{\theta}} \in (x_P^{\dot{\theta}})^{\dot{\theta}}$, что противоречит (23),
  - $k(\tilde{e}) \subseteq k_1^{\dot{\theta}}$, данный случай невозможен по определению термов типа $\mathbf{K}$,
  - $k(\tilde{e}) \subseteq u_1^{\dot{\theta}}$, в данном случае утверждение (13) следует из индуктивного предположения,
- если $f = list$, то $\exists\, i \in 1, \ldots, n$: $k(\tilde{e}) \subseteq e_i^{\dot{\theta}}$, и (13) следует из индуктивного предположения,
- случай $f = shared\_key$ невозможен.

Таким образом, из (24) следует, что $\exists\, x \in x_P^{\dot{\theta}} : k(\tilde{e}) \subseteq x^{\dot{\theta}}$.

Пусть θ'' – первое состояние на пути π, такое, что $(x_P^{\theta''})^{\theta''}$ содержит терм с подтермом $k(\tilde{e})$, т.е.

$$\exists\, x \in x_P^{\theta''} : k(\tilde{e}) \subseteq x^{\theta''}. \tag{25}$$

Из (24) следует, что $\theta'' \leq_\pi \dot{\theta}$. Нетрудно видеть, что θ'' – не начальное состояние, поэтому на пути π существует ребро вида $\ddot{\theta} \xrightarrow{a_{P''}} \theta''$. Обозначим $a = a_{P''}$. Из определения θ'' следует, что $x \notin x_P^{\ddot{\theta}}$, поэтому $P'' = P$ (т.к. при этом переходе изменяется значение x_P), и возможны два случая:

1. $a = \circ?\ddot{e}$, $x \in Var(\ddot{e})$, $\ddot{e}^{\theta''} \in x_\circ^{\ddot{\theta}}$,

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

2. $a = (\ddot{e} := \bar{e})$, $x \in Var(\ddot{e})$, $\bar{e} \in Tm(x_P^{\ddot{\theta}})$, $\ddot{e}^{\theta''} = \bar{e}^{\ddot{\theta}}$,

   поскольку

   • $k(\tilde{e}) \subseteq x^{\theta''} \subseteq \ddot{e}^{\theta''} = \bar{e}^{\ddot{\theta}}$ и

   • согласно (23), $k \notin (x_P^{\dot{\theta}})^{\dot{\theta}}$, поэтому из $\ddot{\theta} <_\pi \dot{\theta}$ следует, что $k \notin (x_P^{\ddot{\theta}})^{\ddot{\theta}}$ (т.к. $x_P^{\ddot{\theta}} \subseteq x_P^{\dot{\theta}}$),

   то, аналогично доказательству импликации (23) $\Rightarrow$ (24) (заменяя в ней $\dot{e}$ на $\bar{e}$ и $\dot{\theta}$ на $\ddot{\theta}$) можно доказать утверждение

   $$\exists\, x \in Var(\bar{e}) \subseteq x_P^{\ddot{\theta}} : k(\tilde{e}) \subseteq x^{\ddot{\theta}},$$

   которое противоречит выбору θ'' как самого первого состояния на пути π со свойством (25): $\ddot{\theta}$ имеет аналогичное свойство, и находится левее θ''. ■

6. Метод верификации протоколов, основанный на представленной модели

6.1. Описание метода верификации

Изложенная в предыдущих пунктах модель криптографических протоколов может применяться для обоснования таких свойств протоколов, которые представляют собой утверждения следующего типа: если при каком-либо выполнении π анализируемого протокола он достиг некоторого состояния $\theta \in \pi$, то существуют состояния $\theta', \ldots \leq_\pi \theta$ на этом выполнении, которые обладают заданными свойствами. В этом пункте в качестве такого свойства рассматривается свойство соответствия в протоколах аутентификации, определяемое ниже. Метод обоснования этого свойства заключается в обратном построении выполнения данного протокола, начиная с состояния θ. Искомые состояния $\theta', \ldots \leq_\pi \theta$ возникают в процессе обратного построения выполнения протокола. Построение данного выполнения производится с использованием теоремы 2.

Ниже излагается иллюстрация применения данного метода для верификации свойств соответствия и секретности протокола аутентификации Yahalom.

6.2. Описание протокола Yahalom

Протокол 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 разделяемый ключ k_{AJ} с доверенным посредником J, на котором A и J могут шифровать и расшифровывать сообщения, используя симметричную систему шифрования.

В каждом сеансе протокола 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. Выполнение сеанса протокола Yahalom с инициатором A, респондером B и доверенным посредником J представляет собой совокупность четырех пересылок сообщений:

$$\begin{array}{lll} 1. & A \to B & : \quad A, n_A \\ 2. & B \to J & : \quad B, k_{BJ}(A, n_A, n_B) \\ 3. & J \to A & : \quad k_{AJ}(B, k, n_A, n_B), k_{BJ}(A, k) \\ 4. & A \to B & : \quad k_{BJ}(A, k), k(n_B) \end{array} \tag{26}$$

Пересылки в (26) 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, B может извлечь из этого сообщения сеансовый ключ k, и
- используя ключ k, B расшифровывает второе сообщение.

Если результат расшифрования совпадает с n_B, то это является для B доказательством того, что отправителем этого сообщения был A.

6.3. Некоторые определения и обозначения

1. Для каждого процесса P запись P^* обозначает РП $\{P_i \mid i \in I\}$, где I – множество натуральных чисел, и $\forall\, i \in I \;\; P_i = P$.

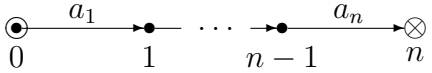
   Будем использовать следующее соглашение:

   - если в каком-либо рассуждении, связанном с РП вида P^*, некоторый процесс является первым из рассматриваемых процессов, входящих в P^*, то этот процесс и все его переменные обозначаются теми же записями, которые используются в P,
   - если кроме этого процесса рассматривается другой процесс, входящий в P^* (возможно совпадающий с P), то он обозначается P_1, и в обозначениях тех его переменных, которые являются дубликатами переменных из множества

   $$Unique_P \cup Private_P \cup \{at_P, x_P\},$$

   используется индекс 1 (например, дубликат переменной x в P_1 будет обозначаться записью x_1), в следущем процессе (P_2, который возможно совпадает с P или P_1) соответствующие переменные будут обозначаться с индексом 2, и т.д.

2. Процесс называется **линейным**, если он имеет вид



(27)

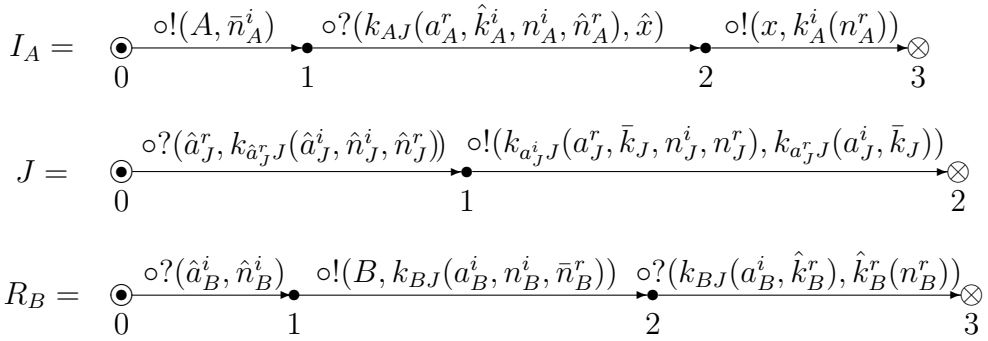
   В целях большей наглядности будем использовать следующее соглашение в обозначениях переменных в линейных процессах: пусть P – процесс вида (27), и переменная x входит в действие a_i, причём $\forall\, j \in \{1, \ldots, i-1\} \;\; x$ не входит в a_j, тогда

   - если $x \in Unique_P$, то будем указывать горизонтальную черту над всеми вхождениями x в a_i (т.е. обозначать их $\bar{x}$) и
   - если $x \in Private_P$, то будем указывать уголок над всеми вхождениями x в a_i (т.е. обозначать их $\hat{x}$).

3. Если π – выполнение РП $\mathcal{P}$, то запись $\pi \ni P^{i,i'} : \theta \xrightarrow{a} \theta'$ имеет следующий смысл: π содержит переход $\theta \xrightarrow{a_P} \theta'$, и $at_P^{\theta} = i, at_P^{\theta'} = i'$.

6.4. Формальное описание процессов, входящих в протокол Yahalom

Описание сеанса протокола Yahalom изображается схемой



В этой схеме первая и третья диаграммы соответствуют процессам I_A и R_B, описывающим поведение инициатора A и респондера B соответственно, вторая диаграмма соответствует процессу, описывающему поведение посредника J, этот процесс обозначается символом J. Верхний индекс i или r при какой-либо переменной означает, что она содержит информацию об инициаторе (i) или респондере (r) данного сеанса. Смысл переменных в этих процессах усматривается из сопоставления действий в этих процессах с соответствующими действиями в (26). Предполагаем, что $Agent_{I_A} = A$, $Agent_{R_B} = B$, $Agent_J = J$.

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

$$\mathcal{P} = \{\{I_A^* \mid A \in Ag\}, \{R_B^* \mid B \in Ag\}, J^*, \dagger\}, \tag{28}$$

т.е. каждый агент может участвовать в неограниченном числе сеансов как в качестве инициатора, так и в качестве респондера.

6.5. Свойства протокола Yahalom

В этом пункте приводится формальное описание и верификация трех свойств протокола (28): секретность ключей k_J и нонсов n_B^r, аутентификация инициатора перед респондером и аутентификация респондера перед инициатором. В доказательствах теорем 3, 4, 5 при каждом применении теоремы 2 имеется единственный вариант обоснования существования перехода (21), и мы будем сразу будем излагать это обоснование, без упоминания о

единственности варианта такого обоснования.

Теорема 3 (секретность ключей k_J и нонсов n^r_B)

РП (28) обладает следующим свойством:

$$\forall\, \theta \in \Theta_{\mathcal{P}}\ \ \theta \models E\bot\dagger, \text{ где } E = \{k_{BJ}, k_J, n^r_B \mid B \in Ag\}. \tag{29}$$

Доказательство.

Докажем (29) от противного.

Предположим, что задано выполнение $\pi = (\theta_0, \theta_1, \ldots)$, и

$$S = \{\theta \in \pi \mid \theta \not\models E\bot\dagger\} \neq \emptyset.$$

Пусть θ – первое состояние на π, которое принадлежит S.

Т.к. $\theta_0 \models E\bot\dagger$, то $\theta \neq \theta_0$, т.е. в π есть переход вида $\theta' \xrightarrow{a_P} \theta$.

Из определения θ следует, что $\theta' \models E\bot\dagger$, $\theta \not\models E\bot\dagger$.

Если бы было верно $P = \dagger$, то, согласно теореме 1, отсюда следовало бы, что $\theta \models E\bot\dagger$, что противоречит определению θ.

Таким образом, $P \in \{I_A, R_B, J \mid A, B \in Ag\}$, и нетрудно видеть, что a_P имеет вид $\circ!e$, $x^\theta_\circ = x^{\theta'}_\circ \cup \{e^{\theta'}\}$, и верно утверждение $\theta \not\models E\bot\dagger$, которое в данной ситуации эквивалентно утверждению

$$\begin{array}{l}\exists\, u \in E,\ \exists \text{ вхождение } u \text{ в } e^{\theta'}, \text{ не содержащееся}\\ \text{ни в каком подтерме вида } k(\ldots) \subseteq e^{\theta'}, \text{ где } k \in E_{\mathbf{K}}.\end{array} \tag{30}$$

Перебором всех возможных обоснований перехода $\theta' \xrightarrow{\circ!e} \theta$ со свойством (30) находим единственное обоснование:

$$\pi \ni I^{2,3}_A : \theta' \overset{\circ!e}{\to} \theta, \text{ где } e = (x, k^i_A(n^r_A)), \tag{31}$$

поэтому (30) можно переписать следующим образом:

$$\begin{array}{l}\exists\, u \in E,\ \exists \text{ вхождение } u \text{ в } e^{\theta'} = (x, k^i_A(n^r_A))^{\theta'},\\ \text{не содержащееся ни в каком подтерме}\\ \text{вида } k(\ldots) \subseteq (x, k^i_A(n^r_A))^{\theta'}, \text{ где } k \in E_{\mathbf{K}}.\end{array} \tag{32}$$

Т.к. $\theta' \vdash at_{I_A} = 2$, то $\exists\, \theta_1 \leq_\pi \theta'$:

$$\pi \ni I^{1,2}_A : \theta'_1 \xrightarrow{\circ?e_1} \theta_1, \text{ где } e_1 = (k_{AJ}(a^r_A, \hat{k}^i_A, n^i_A, \hat{n}^r_A), \hat{x}). \tag{33}$$

Т.к. $\theta'_1 <_\pi \theta_1 \leq_\pi \theta'$ и $\theta' \models E\bot\dagger$, то

$$\theta'_1 \models E\bot\dagger. \tag{34}$$

Из (33) следует, что $e_1^{\theta'_1} = (\ldots, x^{\theta'_1}) \in x_\circ^{\theta'_1}$ (многоточия в терме $(\ldots, x^{\theta'_1})$ и ниже обозначают компоненты, не представляющие интерес для рассмотрения), поэтому, согласно (34) и (7), верно утверждение:

$$\begin{array}{l} \forall\, u \in E \text{ каждое вхождение } u \text{ в } x^{\theta'_1} \subseteq e_1^{\theta'_1} \\ \text{содержится в подтерме } k(\ldots) \subseteq x^{\theta'_1}, \text{ где } k \in E_{\mathbf{K}}. \end{array} \tag{35}$$

Сопоставляя (32) и (35), заключаем:

$$\begin{array}{l} \exists\, u \in E, \exists \text{ вхождение } u \text{ в } (k_A^i(n_A^r))^{\theta'}, \\ \text{не содержащееся ни в каком подтерме} \\ \text{вида } k(\ldots) \subseteq (k_A^i(n_A^r))^{\theta'}, \text{ где } k \in E_{\mathbf{K}}. \end{array} \tag{36}$$

По теореме 2, из (34), $e_1^{\theta'_1} \in x_\circ^{\theta'_1}$ и $k_{AJ} \in E$ следует, что $\exists\, \theta_2 \leq_\pi \theta'_1 : \pi$ содержит переход $\theta'_2 \overset{(\circ!e_2)_P}{\longrightarrow} \theta_2$, где $P \in \mathcal{P}$ и первая компонента $k_{AJ}(\ldots)$ терма $e_1^{\theta'_1}$ входит в $e_2^{\theta'_1}$. Перебором всех вариантов такого перехода находим единственное обоснование:

$$\begin{cases} \pi \ni J^{1,2} : \theta'_2 \overset{\circ!e_2}{\to} \theta_2, \text{ где } e_2 = (k_{a_J^i J}(a_J^r, \bar{k}_J, n_J^i, n_J^r), \ldots) \\ k_{(a_J^i)^\theta J}((a_J^r)^\theta, \bar{k}_J, \ldots) = k_{AJ}(a_A^r, (k_A^i)^\theta, \ldots) \end{cases} \tag{37}$$

Из второй строчки в (37) следует, что $(k_A^i)^\theta = k_J$, поэтому из (36) следует утверждение:

$$\begin{array}{l} \exists\, u \in E, \exists \text{ вхождение } u \text{ в } k_J(\ldots), \text{ не содержащееся} \\ \text{ни в каком подтерме вида } k(\ldots) \subseteq k_J(\ldots), \text{ где } k \in E_{\mathbf{K}}, \end{array}$$

которое, очевидно, противоречиво. ■

Теорема 4 (аутентификация инициатора перед респондером)

РП (28) обладает следующим свойством: $\forall\, R_B \in \mathcal{P}$, $\forall\, \theta \in \Theta_{\mathcal{P}}$, если $\theta \vdash at_{R_B} = 3$, то $\exists\, I_A \in \mathcal{P}$:

$$\theta \vdash \left\{ \begin{array}{l} at_{I_A} = 3 \\ a_A^r = B,\ a_B^i = A \\ n_A^i = n_B^i,\ n_A^r = n_B^r \\ k_A^i = k_B^r \end{array} \right\} \tag{38}$$

Доказательство.

Пусть процесс $R_B \in \mathcal{P}$ и состояние $\theta \in \Theta_{\mathcal{P}}$ таковы, что $\theta \vdash at_{R_B} = 3$, и $\pi = (\theta_0, \theta_1, \ldots)$ – выполнение РП $\mathcal{P}$, такое, что $\theta \in \pi$.

Из $\theta \vdash at_{R_B} = 3$ следует: $\exists\, \theta_1 \leq_\pi \theta$:

$$\pi \ni R_B^{2,3} : \theta_1' \overset{\circ?e_1}{\to} \theta_1, \text{ где } e_1 = (k_{BJ}(a_B^i, \hat{k}_B^r), \hat{k}_B^r(n_B^r)).$$

По теореме (3) верно (29). Согласно теореме (2), в этом случае из $e_1^\theta \in x_\circ^{\theta_1}$ и $k_{BJ} \in E$ следует: $\exists\, \theta_2 \leq_\pi \theta_1'$:

$$\begin{cases} \pi \ni J^{1,2} : \theta_2' \overset{\circ!e_2}{\to} \theta_2, \text{ где } e_2 = (\ldots, k_{a_J^r J}(a_J^i, \bar{k}_J)) \\ k_{(a_J^r)^\theta J}((a_J^i)^\theta, \bar{k}_J) = k_{BJ}((a_B^i)^\theta, (k_B^r)^\theta) \end{cases} \tag{39}$$

Из второй строчки в (39) следует, что

$$(a_J^r)^\theta = B, (a_J^i)^\theta = (a_B^i)^\theta, \bar{k}_J = (k_B^r)^\theta. \tag{40}$$

Из $\theta_2' \vdash at_J = 1$ следует, что $\exists\, \theta_3 \leq_\pi \theta_2'$:

$$\pi \ni J^{0,1} : \theta_3' \overset{\circ?e_3}{\to} \theta_3, \text{ где } e_3 = (\ldots, k_{\hat{a}_J^r J}(\hat{a}_J^i, \hat{n}_J^i, \hat{n}_J^r)). \tag{41}$$

Из (40) и (41) следует, что $k_{BJ}(*,*,*) \subseteq e_3^\theta \in x_\circ^{\theta_3}$ (где звёздочки обозначают некоторые термы), откуда по теореме (2), с учетом соотношений $\theta_3 \models E \perp \dagger$ и $k_{BJ} \in E$ получаем: $\exists\, \theta_4 \leq_\pi \theta_3'$:

$$\begin{cases} \pi \ni R_{B_1}^{1,2} : \theta_4' \overset{\circ!e_4}{\to} \theta_4, \text{ где } e_4 = (\ldots, k_{B_1 J}(a_{B_1}^i, n_{B_1}^i, \bar{n}_{B_1}^r)) \\ k_{B_1 J}((a_{B_1}^i)^\theta, (n_{B_1}^i)^\theta, \bar{n}_{B_1}^r) = k_{BJ}((a_B^i)^\theta, (n_J^i)^\theta, (n_J^r)^\theta) \end{cases} \tag{42}$$

Из второго равенства в (42) следует, что

$$B_1 = B, (n_B^i)^\theta = (n_J^i)^\theta, \bar{n}_B^r = (n_J^r)^\theta. \tag{43}$$

По теореме (2), из $\theta_1 \models E \perp \dagger$, $(k_B^r(n_B^r))^\theta \subseteq e_1^\theta \in x_\circ^{\theta_1}$, и $(k_B^r)^\theta = \bar{k}_J \in E$ следует, что $\exists\, \theta_5 \leq_\pi \theta_1'$:

$$\begin{cases} \pi \ni I_A^{2,3} : \theta_5' \overset{\circ!e_5}{\to} \theta_5, \text{ где } e_5 = (\ldots, k_A^i(n_A^r)) \\ (k_A^i(n_A^r))^\theta = \bar{k}_J(n_B^r) \end{cases} \tag{44}$$

Из второй строчки в (44) следует, что

$$(k_A^i)^\theta = \bar{k}_J, (n_A^r)^\theta = n_B^r. \tag{45}$$

Из $\theta_5 \vdash at_{I_A} = 2$ следует, что $\exists\, \theta_6 \leq_\pi \theta_5'$:

$$\pi \ni I_A^{1,2} : \theta_6' \overset{\circ?e_6}{\to} \theta_6, \text{ где } e_6 = (k_{AJ}(a_A^r, \hat{k}_A^i, n_A^i, \hat{n}_A^r), \ldots). \tag{46}$$

Из (45) и (46) следует, что

$$k_{AJ}(a^r_A,(k^i_A)^\theta,n^i_A,(n^r_A)^\theta)=k_{AJ}(a^r_A,\bar{k}_J,n^i_A,n^r_B)\subseteq e^\theta_6\in x^{\theta_6}_\circ. \tag{47}$$

По теореме (2), из $\theta_6\models E\perp\dagger$, $k_{AJ}\in E$, и (47) следует, что $\exists\,\theta_7\leq_\pi\theta'_6$:

$$\begin{cases}\pi\ni J^{1,2}_1:\theta'_7\overset{\circ!e_7}{\to}\theta_7,\ \text{где}\ e_7=(k_{a^i_{J_1}J_1}(a^r_{J_1},\bar{k}_{J_1},n^i_{J_1},n^r_{J_1}),\ldots)\\ k_{(a^i_{J_1})^\theta J}((a^r_{J_1})^\theta,\bar{k}_{J_1},(n^i_{J_1})^\theta,(n^r_{J_1})^\theta)=k_{AJ}(a^r_A,\bar{k}_J,n^i_A,n^r_B)\end{cases} \tag{48}$$

Из второй строчки в (48) следует, что

$$(a^i_{J_1})^\theta=A,(a^r_{J_1})^\theta=a^r_A,J_1=J,(n^i_J)^\theta=n^i_A,(n^r_J)^\theta=\bar{n}^r_B. \tag{49}$$

Свойство (38) следует из (40), (43), (45), (49). ■

Теорема 5 (аутентификация респондера перед инициатором)

РП (28) обладает следующим свойством: $\forall\,I_A\in\mathcal{P},\forall\,\theta\in\Theta_\mathcal{P}$, если $\theta\vdash at_{I_A}=2$, то $\exists\,R_B\in\mathcal{P}$:

$$\theta\vdash\left\{\begin{array}{l}at_{R_B}=2,\\ a^r_A=B,a^i_B=A,\\ n^i_A=n^i_B,n^r_A=n^r_B\end{array}\right\}. \tag{50}$$

Доказательство.

Пусть процесс I_A и состояние θ таковы, что $\theta\vdash at_{I_A}=2$, и $\pi=(\theta_0,\theta_1,\ldots)$ – выполнение РП $\mathcal{P}$, такое, что $\theta\in\pi$.

Из $\theta\vdash at_{I_A}=2$ следует, что $\exists\,\theta_1\leq_\pi\theta$:

$$\pi\ni I^{1,2}_A:\theta'_1\overset{\circ?e_1}{\to}\theta_1,\ \text{где}\ e_1=(k_{AJ}(a^r_A,\hat{k}^i_A,n^i_A,\hat{n}^r_A),\ldots). \tag{51}$$

По теореме 2, из $\theta_1\models E\perp\dagger$, $k_{AJ}(*,*,*,*)\subseteq e^\theta_1\in x^{\theta_1}_\circ$ (где звёздочки обозначают некоторые термы) и $k_{AJ}\in E$, следует, что $\exists\,\theta_2\leq_\pi\theta'_1$:

$$\begin{cases}\pi\ni J^{1,2}:\theta'_2\overset{\circ!e_2}{\to}\theta_2,\ \text{где}\ e_2=(k_{a^i_JJ}(a^r_J,\bar{k}_J,n^i_J,n^r_J),\ldots)\\ k_{(a^i_J)^\theta J}((a^r_J)^\theta,\bar{k}_J,(n^i_J)^\theta,(n^r_J)^\theta)=k_{AJ}(a^r_A,(k^i_A)^\theta,n^i_A,(n^r_A)^\theta)\end{cases} \tag{52}$$

Из второй строчки в (52) следует, что

$$\begin{aligned}&(a^i_J)^\theta=A,(a^r_J)^\theta=a^r_A,\bar{k}_J=(k^i_A)^\theta,\\ &(n^i_J)^\theta=n^i_A,(n^r_J)^\theta=(n^r_A)^\theta.\end{aligned} \tag{53}$$

Из $\theta_2 \vdash at_J = 1$ следует, что $\exists\, \theta_3 \leq_\pi \theta'_2$:

$$\pi \ni J^{0,1} : \theta'_3 \overset{\circ?e_3}{\to} \theta_3, \text{ где } e_3 = (\ldots, k_{\hat{a}^r_J J}(\hat{a}^i_J, \hat{n}^i_J, \hat{n}^r_J)). \tag{54}$$

Из (53) и (54) следует, что $k_{a^r_A J}(A, n^i_A, (n^r_A)^\theta) \subseteq e_3^\theta \in x_\circ^{\theta_3}$, откуда по теореме (2), учитывая $\theta_3 \models E \perp \dagger$, и $k_{a^r_A J} \in E$, получаем: $\exists\, \theta_4 \leq_\pi \theta'_3$:

$$\begin{cases} \pi \ni R^{1,2}_B : \theta'_4 \overset{\circ!e_4}{\to} \theta_4, \text{ где } e_4 = (\ldots, k_{BJ}(a^i_B, n^i_B, \bar{n}^r_B)) \\ k_{BJ}((a^i_B)^\theta, (n^i_B)^\theta, \bar{n}^r_B) = k_{a^r_A J}(A, n^i_A, (n^r_A)^\theta). \end{cases} \tag{55}$$

Второе равенство в (55) влечёт равенства, из которых следует (50):

$$B = a^r_A, (a^i_B)^\theta = A, (n^i_B)^\theta = n^i_A, \bar{n}^r_B = (n^r_A)^\theta. \quad \blacksquare$$

7. Верификация протокола передачи 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 известного КП Wide-Mouth Frog и изложен в работе [96].

7.1. 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:

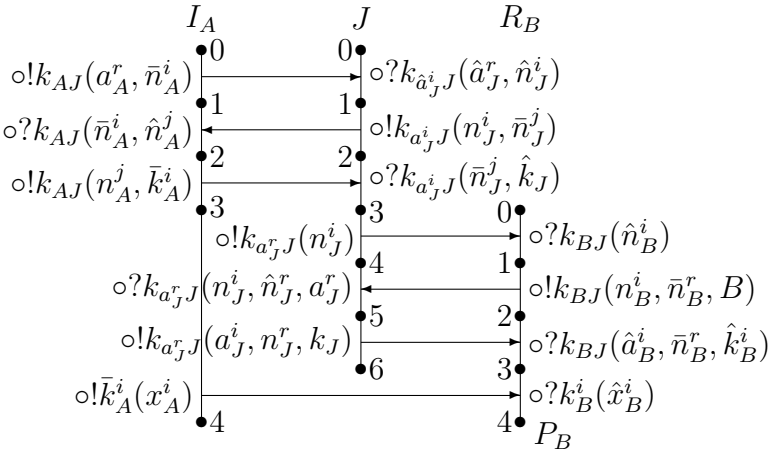
- обмен сообщениями между A и J, в результате чего J узнает имя A отправителя, имя B получателя, и ключ k, на котором будет зашифровано сообщение x от A для получателя B,
- обмен сообщениями между J и B, в результате чего B узнает имя A отправителя сообщения, которое B получит от A, и ключ k, на котором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

Выполнение сеанса данного КП с инициатором A, респондером B и доверенным по-

средником J представляет собой следующую совокупность пересылок сообщений:

$$\begin{array}{lll} 1. & A \to J & : \ k_{AJ}(A, n_A) \\ 2. & J \to A & : \ k_{AJ}(n_A, n_J) \\ 3. & A \to J & : \ k_{AJ}(n_J, k) \\ 4. & J \to B & : \ k_{BJ}(n_A) \\ 5. & B \to J & : \ k_{BJ}(n_A, n_B, B) \\ 6. & J \to B & : \ k_{BJ}(A, n_B, k) \\ 7. & A \to B & : \ k(x) \end{array} \tag{56}$$

Данный сеанс представляется следующей схемой:



(57)

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

$$\mathcal{P} = \{\{I_A^* \mid A \in Ag\}, \{R_B^* \mid B \in Ag\}, J^*, \dagger\}. \tag{58}$$

Свойства этого КП, которые должны быть верифицированы:

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

$$\forall\, c \in \Theta_{\mathcal{P}} \ \ \theta \models E \perp \dagger, \ \ \text{где } E = \{k_{AJ}, k_A^i, x_A^i, n_A^i \mid A \in Ag\} \tag{59}$$

• **целостность** передаваемых сообщений:

$$\begin{array}{l} \forall\, R_B \in \mathcal{P}, \forall\, \theta \in \Theta_{\mathcal{P}}, \text{если } \theta \vdash at_{R_B} = 4, \text{ то } \exists\ I_A \in \mathcal{P}: \\ \theta \vdash \{at_{I_A} = 4, a_A^r = B, a_B^i = A, n_A^i = n_B^i, k_A^i = k_B^i, x_A^i = x_B^i\} \end{array} \tag{60}$$

7.2. Верификация протокола

Доказательство свойства секретности (59) дословно повторяет начало рассуждений по доказательству аналогичного свойства протокола Yahalom.

Докажем свойство целостности (60). Будем использовать в этом доказательстве свойство (59) (не упоминая об этом).

Пусть процесс $R_B \in \mathcal{P}$ и состояние $\theta \in \Theta_{\mathcal{P}}$ таковы, что $\theta \vdash at_{R_B} = 4$. Докажем, что $\exists\, I_A \in \mathcal{P}$: выполнено утверждение во второй строчке (60).

Пусть π – путь из θ^0 в θ. Из $\theta \vdash at_{R_B} = 4$ следует, что

$$\begin{array}{l} \exists\, \theta_1 \leq_\pi \theta :\ \pi \ni R_B^{3,4} : \theta_1' \overset{\circ ? e_1}{\to} \theta_1,\ \text{где } e_1 = k_B^i(\hat{x}_B^i) \\ \exists\, \theta_2 \leq_\pi \theta_1' :\ \pi \ni R_B^{2,3} : \theta_2' \overset{\circ ? e_2}{\to} \theta_2,\ \text{где } e_2 = k_{BJ}(\hat{a}_B^i, \bar{n}_B^r, \hat{k}_B^i) \end{array} \tag{61}$$

По теореме 2, из второй строки в (61), $e_2^\theta \in x_\circ^{\theta_2}$, $k_{BJ} \in E$, следует:

$$\begin{cases} \exists\, \theta_3 \leq_\pi \theta_2' : \pi \ni J^{5,6} : \theta_3' \overset{\circ ! e_3}{\to} \theta_3,\ \text{где } e_3 = k_{a_J^r J}(a_J^i, n_J^r, k_J) \\ k_{(a_J^r)^\theta J}((a_J^i)^\theta, (n_J^r)^\theta, (k_J)^\theta) = k_{BJ}(\hat{a}_B^i, \bar{n}_B^r, \hat{k}_B^i) \end{cases} \tag{62}$$

Из второй строки в (62) следует, что

$$(a_J^r)^\theta = B,\ (a_J^i)^\theta = (a_B^i)^\theta,\ (n_J^r)^\theta = \bar{n}_B^r,\ (k_J)^\theta = (k_B^i)^\theta. \tag{63}$$

Из первой строки в (62), с учетом (63), получаем:

$$\exists\, \theta_4 \leq_\pi \theta_3' : \pi \ni J^{4,5} : \theta_4' \overset{\circ ? e_4}{\to} \theta_4,\ \text{где } e_4 = k_{BJ}(n_J^i, n_B^r, B). \tag{64}$$

По теореме 2, из (64), и того, что $e_4^\theta \in x_\circ^{\theta_4}$, $k_{BJ} \in E$, следует:

$$\begin{cases} \exists\, \theta_5 \leq_\pi \theta_4' : \pi \ni B_1^{1,2} : \theta_5' \overset{\circ ! e_5}{\to} \theta_5,\ \text{где } e_5 = k_{B_1 J}(n_{B_1}^i, \bar{n}_{B_1}^r, B_1) \\ k_{B_1 J}((n_{B_1}^i)^\theta, \bar{n}_{B_1}^r, B_1) = k_{BJ}((n_J^i)^\theta, \bar{n}_B^r, B) \end{cases} \tag{65}$$

Из второй строки в (65) следует, что

$$\bar{n}_{B_1}^r = \bar{n}_B^r,\ B_1 = B,\ (n_B^i)^\theta = (n_J^i)^\theta. \tag{66}$$

Из (64) следует, что

$$\exists\, \theta_6 \leq_\pi \theta_4' : \pi \ni J^{2,3} : \theta_6' \overset{\circ ? e_6}{\to} \theta_6,\ \text{где } e_6 = k_{a_J^i J}(n_J^j, k_J). \tag{67}$$

По теореме 2, из (67), и того, что $e_6^\theta \in x_\circ^{\theta_6}$, $k_{(a_J^i)^\theta J} \in E$, следует:

$$\begin{cases} \exists\, \theta_7 \leq_\pi \theta_6' : \pi \ni A^{2,3} : \theta_7' \overset{\circ ! e_7}{\to} \theta_7,\ \text{где } e_7 = k_{AJ}(n_A^j, \bar{k}_A^i) \\ k_{AJ}((n_A^j)^\theta, \bar{k}_A^i) = k_{(a_J^i)^\theta J}(\bar{n}_J^j, (k_J)^\theta) \end{cases} \tag{68}$$

Из второй строки в (68) получаем:

$$A = (a_J^i)^\theta,\ (n_A^j)^\theta = \bar{n}_J^j,\ \bar{k}_A^i = (k_J)^\theta \tag{69}$$

Из первой строки в (68) получаем:

$$\exists\, \theta_8 \leq_\pi \theta'_7 : \pi \ni I_A^{1,2} : \theta'_8 \overset{\circ?e_8}{\to} \theta_8, \text{ где } e_8 = k_{AJ}(\bar{n}^i_A, \hat{n}^j_A). \tag{70}$$

По теореме 2, из (70), и того, что $e_8^\theta \in x_\circ^{\theta_8}$, $k_{AJ} \in E$, следует:

$$\begin{cases} \exists\, \theta_9 \leq_\pi \theta'_8 : \pi \ni {J_1}^{1,2} : \theta'_9 \overset{\circ!e_9}{\to} \theta_9, \text{ где } e_9 = k_{a^i_{J_1} J}(n^i_{J_1}, \bar{n}^j_{J_1}) \\ k_{(a^i_{J_1})^\theta J}((n^i_{J_1})^\theta, \bar{n}^j_{J_1}) = k_{AJ}(\bar{n}^i_A, (n^j_A)^\theta) \end{cases} \tag{71}$$

Из второй строки в (71) получаем:

$$(a^i_{J_1})^\theta = A,\ (n^i_{J_1})^\theta = \bar{n}^i_A,\ \bar{n}^j_{J_1} = (n^j_A)^\theta \tag{72}$$

Из (69) и (72) получаем:

$$\bar{n}^j_{J_1} = \bar{n}^j_J = (n^j_A)^\theta,\ J_1 = J,\ (a^i_J)^\theta = A,\ (n^i_J)^\theta = \bar{n}^i_A. \tag{73}$$

Из (63) и (69) получаем:

$$(k^i_B)^\theta = (k_J)^\theta = \bar{k}^i_A \in E, \tag{74}$$

поэтому по теореме 2, из первой строки в (61) и $e_1^\theta \in x_\circ^{\theta_1}$ следует:

$$\begin{cases} \exists\, \theta_{11} \leq_\pi \theta'_1 : \pi \ni {A_1}^{3,4} : \theta'_{11} \overset{\circ!e_{11}}{\to} \theta_{11}, \text{ где } e_{11} = \bar{k}^i_{A_1}(x^i_{A_1}) \\ \bar{k}^i_{A_1}(x^i_{A_1}) = (k^i_B)^\theta((x^i_B)^\theta) \end{cases} \tag{75}$$

Из второй строки в (75) и (74) получаем:

$$\bar{k}^i_{A_1} = (k^i_B)^\theta = \bar{k}^i_A,\ A_1 = A,\ x^i_A = (x^i_B)^\theta. \tag{76}$$

Из первой строки в (71) и (73) получаем:

$$\exists\, \theta_{10} \leq_\pi \theta'_9 : \pi \ni J^{0,1} : {\theta'_{10}}^{\circ?e_{10}} \to \theta_{10}, \text{ где } e_{10} = k_{\hat{a}^i_J J}(\hat{a}^r_J, \hat{n}^i_J). \tag{77}$$

Из (63) и (73) следует, что $e_{10}^\theta = k_{AJ}(B, \bar{n}^i_A)$.

По теореме 2, из (77), $e_{10}^\theta \in x_\circ^{\theta_{10}}$, $k_{AJ} \in E$, следует:

$$\begin{cases} \exists\, \theta_{12} \leq_\pi \theta'_{10} : \pi \ni {A_1}^{0,1} : {\theta'_{12}}^{\circ!e_{12}} \to \theta_{12}, \text{ где } e_{12} = k_{A_1 J}(a^r_{A_1}, \bar{n}^i_{A_1}) \\ k_{A_1 J}(a^r_{A_1}, \bar{n}^i_{A_1}) = k_{AJ}(B, \bar{n}^i_A) \end{cases} \tag{78}$$

Из второй строки в (78) получаем:

$$\bar{n}^i_{A_1} = \bar{n}^i_A,\ A_1 = A,\ a^r_A = B. \tag{79}$$

Утверждение (60) обосновывается следующим образом:

- $\theta \vdash at_{I_A} = 4$ следует из (75), (76): $\theta_{11} \vdash at_{A_1} = 4$, $A_1 = A$, $\theta_{11} \leq_\pi s$,
- $\theta \vdash a^r_A = B$ следует из (79),
- $\theta \vdash a^i_B = A$ следует из (63) и (69),
- $\theta \vdash n^i_A = n^i_B$ следует из (66) и (73),
- $\theta \vdash k^i_A = k^i_B$ следует из (74),
- $\theta \vdash x^i_A = x^i_B$ следует из (76). ■

8. Заключение

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

1. Кондратьев Д.А., Бояндин Л.К., Гончар Г.Е., Марченко В.В., Обухова А.А., Разбитнова Ю.Ю., Хованская А.С., Янбулатов Д.Р. Формальная верификация реализации хэш-функции «Стрибог» с «Группой Астра», Системная информатика, 2025. – № 28, – С. 25-52.
2. Миронов А.М., Математическая модель и методы верификации криптографических протоколов, Интеллектуальные системы. Теория и приложения, издательство ООО "Интеллектуальные системы"(Москва), том 26, № 2, с. 85-144, 2022.
3. Миронов А.М. Методы верификации программ. ДМК Пресс Москва, 2023, 335 с.
4. 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.
5. M. Abadi, B. Blanchet, C. Fournet. Just Fast Keying in the Pi Calculus. In ACM

Transactions on Information and System Security, 10(3), 2007.

6. M. Abadi, B. Blanchet, C. Fournet. The Applied Pi Calculus: Mobile Values, New Names, and Secure Communication. Journal of the ACM, Volume 65, Issue 1 Article No.: 1, Pages 1 - 41, 2017.
7. M. Abadi, A.D. Gordon, A calculus for cryptographic protocols: the Spi calculus. Inf. Comput. 148, issue 1, 1–70 (1999)
8. 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
9. M. Abadi, M.R. Tuttle, A semantics for a logic of authentication (extended abstract), in 10th ACM Symposium on Principles of Distributed Computing (PODC'91), Montreal, Canada (ACM, New York, 1991), pp. 201–216
10. R. Anderson and R. Needham. Programming Satan's computer. In J. van Leeuwen (ed.) Computer Science Today, volume 1000 of LNCS. Springer, 1995, pp. 426–440.
11. Andrew W. Appel, Verification of a Cryptographic Primitive: SHA-256, ACM Transactions on Programming Languages and Systems (TOPLAS), Volume 37, Issue 2, p. 1-31, 2015.
12. G. Bella. Inductive Verification of Cryptographic Protocols. PhD thesis, Cambridge University, 2000.
13. B. Blanchet. An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In 14th IEEE Computer Security Foundations Workshop (CSFW), pp. 82-96, 2001.
14. Bruno Blanchet, Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif, Foundations and Trends in Privacy and Security: Vol. 1: No. 1-2, pp 1-135, 2016.
15. Burrows M., Abadi M., Needham R., A Logic of Authentication. In ACM Transactions on Computer Systems, 8(1), (1990) 18-36.
16. I. Cervesato, N.A. Durgin, P.D. Lincoln, J.C. 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
17. 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.
18. Véronique Cortier, Stephanie Delaune, and Vaishnavi Sundararajan. A Decidable Class of Security Protocols for Both Reachability and Equivalence Properties. Journal of

Automated Reasoning, vol. 65, issue 4, 479–520, April 2021.

19. 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.
20. Véronique Cortier, Steve Kremer. Formal Models and Techniques for Analyzing Security Protocols: A Tutorial. Foundations and Trends in Programming Languages, 1(3):151–267, (2014)
21. Véronique Cortier and Cyrille Wiedling. A formal analysis of the Norwegian E-voting protocol. Journal of Computer Security, 25(1):21–57, 2016.
22. 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.
23. Cas Cremers, Sjouke Mauw. Operational Semantics and Verification of Security Protocols, Springer-Verlag Berlin Heidelberg, 2012.
24. 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.
25. A. Datta, J.C. Mitchell, A. Roy, S.H. Stiller, Protocol composition logic, in Formal Models and Techniques for Analyzing Security Protocols, ed. by V. Cortier, S. Kremer (IOS Press, Lansdale, 2011)
26. Denning D.E., Sacco G., Timestamps in Key Distribution Protocols, Communications of the ACM, Vol. 24, No. 8, (1981) 533-536.
27. 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
28. 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)
29. 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

30. 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. 241–255.
31. B. Dutertre and S. A. Schneider. Using a PVS embedding of CSP to verify authentication protocols. Number 1275 in LNCS. Springer, 1997, pp. 121-136.
32. L. Gong, R.M. Needham, R. Yahalom, Reasoning about belief in cryptographic protocols, in 1990 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, USA (IEEE Computer Society, Los Alamitos, 1990), pp. 234–248
33. Joshua D. Guttman. State and Progress in Strand Spaces: Proving Fair Exchange. Journal of Automated Reasoning, 48(2): 159-195, 2012.
34. J. D. Guttman and F. J. Thayer. Authentication tests. IEEE Computer Society Symposium on Research in Security and Privacy, 2000, pp. 96-109.
35. J.D. Guttman, F.J. Thayer, Authentication tests and the structure of bundles. Theor. Comput. Sci. 283(2), 333–380 (2002)
36. C. A. R. Hoare. Communicating Sequential Processes. Prentice-Hall, 1985, 256 pages.
37. R.A. Kemmerer, C. Meadows, J.K. Millen, Three systems for cryptographic protocol analysis. J. Cryptol. 7, 79–130 (1994)
38. 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.
39. Yongjian Li, Jun Pang. An inductive approach to strand spaces. Formal Aspects of Computing, Vol. 25, No. 4, 2013, pp. 465–501.
40. Gavin Lowe. An attack on the Needham-Schroeder public key authentication protocol. Information Processing Letters, 56(3):131–133, November 1995.
41. G. Lowe and B. Roscoe. Using CSP to detect errors in the TMN protocol. IEEE Transactions in Software Engineering, 23(10), 1997, pp. 659 - 669.
42. R. Milner, A Calculus of Communicating Systems, Springer Verlag, 1980, 176 p.
43. Needham R.M., Schroeder M.D., Authentication revisited, ACM SIGOPS Operating Systems Review, Vol. 21, No. 1, p. 7 (1987).
44. Roger Needham and Michael Schroeder. Using encryption for authentification in large networks of computers. Communications of the ACM, 21(12), December 1978, pp. 993-999.

45. 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
46. L. C. Paulson. Inductive Analysis of the Internet Protocol TLS. In ACM Trans. on Information and System Security, 2(3), pp. 332-351, 1999.
47. L.C. Paulson, The inductive approach to verifying cryptographic protocols. J. Comput. Secur. 6(1–2), 85–128 (1998)
48. 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.
49. Roggenbach, M., Cerone, A., Schlingloff, B.- 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, 2022.
50. P. Y. A. Ryan and S. A. Schneider. Process algebra and non-interference. Journal of Computer Security, vol. 9, issue 1-2, 2001, pp. 75-103.
51. 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, 320 pages.
52. Mark D. Ryan and Ben Smyth, Applied pi calculus, in: Formal Models and Techniques for Analyzing Security Protocols, Edited by Véronique Cortier and Steve Kremer, 2011 IOS Press, p. 112-142.
53. S. A. Schneider. Verifying authentication protocols in CSP. IEEE Transactions on Software Engineering, vol. 24, issue 9, 1998, pp. 741-758.
54. 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.
55. S. A. Schneider and A. Sidiropoulos. CSP and anonymity. ESORICS '96: Proceedings of the 4th European Symposium on Research in Computer Security: Computer Security Pages 198 - 218
56. 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)
57. Syverson P., Meadows C., A Logical Language for Specifying Cryptographic Protocol

Requirements, Proceedings of the 1993 IEEE Computer Society Symposium on Research in Security and Privacy, (1993) 165-177, IEEE Computer Society Press.

58. P.F. Syverson, P.C. van Oorschot, A unified cryptographic protocol logic. CHACS Report 5540-227 NRL (1996)
59. F.J. Thayer Fábrega, 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
60. F.J. Thayer Fábrega, 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
61. F.J. Thayer Fábrega, J. C. Herzog, and J. D. Guttman. Strand spaces: Proving security protocols correct. Journal of Computer Security, 7(2/3):191-230, 1999.
62. F.J. Thayer Fábrega, J. C. Herzog, and J. D. Guttman. Strand spaces: why is a security protocol correct? 1998 IEEE Symposium on Security and Privacy, 1998, pp. 160-171.
63. Fan Yang, Santiago Escobar, Catherine A Meadows, José 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.
64. Véronique 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.
65. 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.
66. Véronique Cortier and Cyrille Wiedling. A formal analysis of the Norwegian E-voting protocol. Journal of Computer Security, 25(15777):21–57, 2017.
67. 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`
68. Bruno Blanchet, Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif, 2016.

69. Fan Yang, Santiago Escobar, Catherine A Meadows, José 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.
70. Véronique Cortier, Steve Kremer. Formal Models and Techniques for Analyzing Security Protocols: A Tutorial. Foundations and Trends in Programming Languages, 1(3):151–267, (2014)
71. Yongjian Li, Jun Pang. An inductive approach to strand spaces. Formal Aspects of Computing, Vol. 25, No. 4, 2013.
72. Cas Cremers, Sjouke Mauw. Operational Semantics and Verification of Security Protocols, Springer-Verlag Berlin Heidelberg, 2012.
73. Joshua D. Guttman. State and Progress in Strand Spaces: Proving Fair Exchange. Journal of Automated Reasoning, 48(2): 159-195, 2012.
74. 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.
75. 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)
76. Mark D. Ryan and Ben Smyth, Applied pi calculus, in: Formal Models and Techniques for Analyzing Security Protocols, Edited by Véronique Cortier, 2011 IOS Press, p. 112-142.
77. 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
78. 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.
79. M. Abadi, B. Blanchet, C. Fournet, Just Fast Keying in the Pi Calculus. In ACM Transactions on Information and System Security, 10(3), 2007.
80. 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
81. 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)

82. 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
83. 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
84. 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.
85. I. Cervesato, N.A. Durgin, P.D. Lincoln, J.C. 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
86. 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.
87. J.D. Guttman, F.J. Thayer, Authentication tests and the structure of bundles. Theor. Comput. Sci. 283(2), 333–380 (2002)
88. 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)
89. 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
90. B. Blanchet. An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In 14th IEEE Computer Security Foundations Workshop (CSFW), pp. 82-96, 2001.
91. 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
92. G. Bella. Inductive Verification of Cryptographic Protocols. PhD thesis, Cambridge University, 2000.
93. 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.

94. 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.
95. P. Y. A. Ryan and S. A. Schneider. Process algebra and non-interference. Journal of Computer Security, 2000.
96. M. Abadi, A.D. Gordon, A calculus for cryptographic protocols: the Spi calculus. Inf. Comput. 148, 1–70 (1999)
97. L. C. Paulson. Inductive Analysis of the Internet Protocol TLS. In ACM Trans. on Information and System Security, 2(3), pp. 332-351, 1999.
98. 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.
99. 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
100. L.C. Paulson, The inductive approach to verifying cryptographic protocols. J. Comput. Secur. 6(1–2), 85–128 (1998)
101. 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
102. 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.
103. S. A. Schneider. Verifying authentication protocols in CSP. IEEE Transactions on Software Engineering, 1998.
104. 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.
105. G. Lowe and A. W. Roscoe. Using CSP to detect errors in the TMN protocol. IEEE Transactions in Software Engineering, 23(10), 1997.
106. 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
107. 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.

108. P.F. Syverson, P.C. van Oorschot, A unified cryptographic protocol logic. CHACS Report 5540-227 NRL (1996)
109. S. A. Schneider and A. Sidiropoulos. CSP and anonymity. European Symposium on Research in Computer Security, 1996.
110. R. Anderson and R. Needham. Programming Satan's computer. In J. van Leeuwen (ed.) Computer Science Today, volume 1000 of LNCS. Springer, 1995.
111. Gavin Lowe. An attack on the Needham-Schroeder public key authentication protocol. Information Processing Letters, 56(3):131–136, November 1995.
112. R.A. Kemmerer, C. Meadows, J.K. Millen, Three systems for cryptographic protocol analysis. J. Cryptol. 7, 79–130 (1994)
113. 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
114. 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.
115. 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
116. Burrows M., Abadi M., Needham R., A Logic of Authentication. In ACM Transactions on Computer Systems, 8(1), (1990) 18-36.
117. 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
118. Needham R., Schroeder M., Authentication revisited, Operating Systems Review, Vol. 21, No. 1, (1987).
119. C. A. R. Hoare. Communicating Sequential Processes. Prentice-Hall, 1985.
120. Denning D., Sacco G., Timestamps in Key Distribution Protocols, Communications of the ACM, Vol. 24, No. 8, (1981) 533-536.
121. R. Milner, A Calculus of Communicating Systems, Springer Verlag, 1980.
122. Roger Needham and Michael Schroeder. Using encryption for authentification in large

networks of computers. Communications of the ACM, 21(12), December 1978.

123. M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In POPL'01: Proceedings of the 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, pages 104–115. ACM Press, 2001.
124. M. Abadi, B. Blanchet, and C. Fournet. 2017. The Applied Pi Calculus: Mobile Values, New Names, and Secure Communication. J. ACM 65, 1, Article 1 (October 2017), 103 pages.
125. Миронов А.М. Методы верификации программ. ДМК Пресс Москва, 2023, 335 с.