Процессный подход к верификации криптографических протоколов

Андрей Михайлович Миронов

amironov66@gmail.com

Московский Государственный Университет
механико-математический факультет

Software Engineering, Theory and Experimental Programming
11 октября 2024 г.

Описание доклада

В докладе будет изложена новая математическая модель криптографических протоколов, и приводятся примеры применения этой модели для решения задач верификации криптографических протоколов. Модель криптографических протоколов основана на понятии последовательного и распределенного процессов.
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 настоящем докладе, 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.

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.

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, 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, 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: 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, 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.
- уязвимости 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.

Последовательные и распределенные процессы

Последовательный процесс является моделью участника КП, а распределенный процесс является моделью всего КП.

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.

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 $Var_{\mathbf{A}}$ – 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).

Процесс противника

Процесс противника – это процесс $\dagger$, обладающий свойствами:

- граф процесса $\dagger$ состоит из единственной вершины,
- $\forall\, a \in Act$ граф процесса $\dagger$ содержит ребро с меткой a.

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

Распределенные процессы

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

$$\{Private_{P_i} \cup Unique_{P_i} \cup \{x_{P_i}, at_{P_i}\} \mid i \in I\} \tag{1}$$

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

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

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

1. $at_P^\theta = v$, $at_P^{\theta'} = v'$, $\forall\, x \in x_P^\theta \setminus \{at_P, x_P, x_\circ\}\ \ x^\theta = x^{\theta'}$,
2. 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

$$\text{либо } k \in Var, \text{ либо } \left\{ \begin{array}{l} k = shared\_key(\ldots) \\ Agent_P \in k \end{array} \right\}, \tag{2}$$

3. если $a = \circ?e$ или $e := e'$, то
   1. $x_\circ^{\theta'} = x_\circ^\theta$, $x_P^{\theta'} = x_P^\theta \cup Var(e)$,
   2. $\theta' \vdash e \in x_\circ$ или $\left\{ \begin{array}{l} \theta' \vdash e = e' \\ e' \in Tm(x_P^\theta) \end{array} \right\}$ соответственно,
   3. если e^θ содержит подтерм вида $k(\tilde{e})$, где $k \in Tm_{\mathbf{K}}$, то верно (2),
   4. если $a = (e := e')$ и $k = shared\_key(\ldots)$, то верна импликация

$$k \subseteq (e')^\theta \ \Rightarrow\ Agent_P \in k. \tag{3}$$

Интерпретация переходов

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.

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 $\theta_i \xrightarrow{a_P} \theta_{i+1}$, 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 $\mathcal{P}$.

Свойство защищённости

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

$$E \perp P, \text{ где } \begin{cases} E \subseteq Public_{\mathcal{P}} \cup Tm(Public_{\mathcal{P}})_{\mathbf{K}}, \\ P \in \mathcal{P}, \forall\, k \in E_{\mathbf{K}} \;\; Agent_P \notin k, \end{cases} \tag{4}$$

где $\mathcal{P}$ – некоторый РП.
(Напомним, что $E_{\mathbf{K}}$ – это множество термов из E типа $\mathbf{K}$ (ключ))
Свойство (4) истинно в состоянии $\theta \in \Theta_{\mathcal{P}}$ (что обозначается записью $\theta \models E \perp P$), если

$$\begin{gathered} \forall\, e \in E,\ \forall\, e' \in (x_P^\theta)^\theta \cup x_\circ^\theta \text{ каждое вхождение } e \text{ в } e' \\ \text{содержится в подтерме } k(\ldots) \subseteq e',\ \text{где } k \in E_{\mathbf{K}}. \end{gathered} \tag{5}$$

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.

Сохранение защищённости при переходах

Ниже приводится 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

$$\theta \models E \perp P \Rightarrow \theta' \models E \perp P. \tag{6}$$

Свойство соответствия

Ниже приводится 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.
Данная 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 $\theta \models E \perp P$, и $\exists\, e \in x_{\circ}^{\theta}$:

$$\exists\, k(\tilde{e}) \subseteq e, \text{ где } k \in E_{\mathbf{K}}. \tag{7}$$

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

$$\dot{\theta} \xrightarrow{(\circ ! \dot{e})_{P'}} \theta', \text{ где } k(\tilde{e}) \subseteq \dot{e}^{\dot{\theta}} \text{ и } \theta' \leq_{\pi} \theta. \tag{8}$$

Метод верификации протоколов, основанный на представленной модели

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.

Ниже 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.

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:

$$\begin{array}{lrcl} 1. & A \to B & : & A, n_A \\ 2. & B \to J & : & B, k_{BJ}(A, n_A, n_B) \\ 3. & J \to A & : & k_{AJ}(B, k, n_A, n_B), k_{BJ}(A, k) \\ 4. & A \to B & : & k_{BJ}(A, k), k(n_B) \end{array} \tag{9}$$

Пересылки в (9) 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.

Некоторые определения и обозначения

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

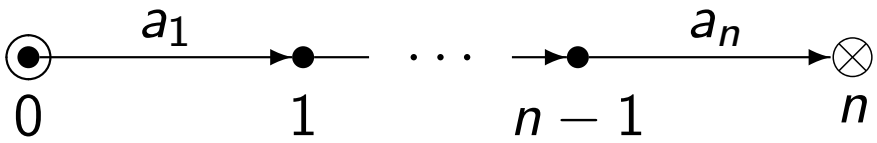
Будем использовать следующее соглашение:

- 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.

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



(10)

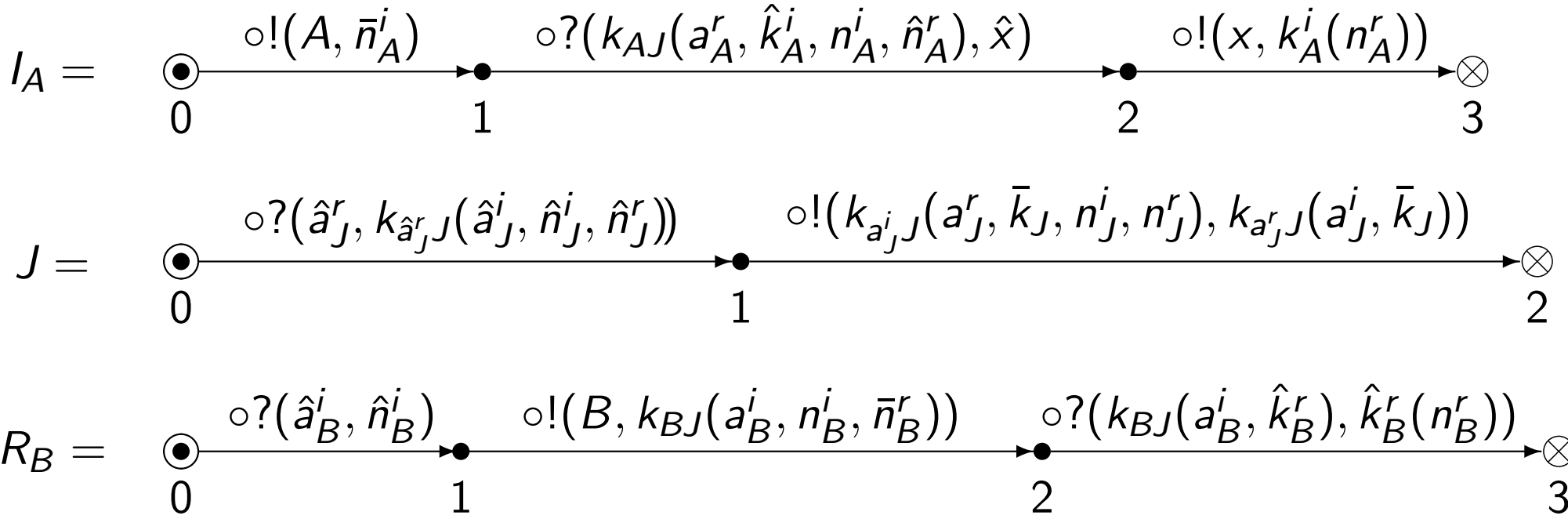
В целях большей наглядности будем использовать следующее соглашение в обозначениях переменных в линейных процессах: пусть P – процесс вида (10), и переменная x входит в действие a_i, причём $\forall\, j \in \{1, \ldots, i-1\}$ x не входит в a_j, тогда

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

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 $at_P^{\theta} = i, at_P^{\theta'} = i'$.

Формальное описание процессов, входящих в 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 (9).
Предполагаем, что $Agent(I_A) = A$, $Agent(R_B) = B$, $Agent(J) = J$.

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

$$\mathcal{P} = \{\{I_A^* \mid A \in Ag\}, \{R_B^* \mid B \in Ag\}, J^*, \dagger\}, \tag{11}$$

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

Свойства протокола Yahalom

Ниже приводится формальное описание и верификация трех свойств протокола (11):

- секретность ключей k_J и нонсов n_B^r,
- аутентификация инициатора перед респондером и
- аутентификация респондера перед инициатором.

секретность ключей k_J и нонсов n_B^r

Теорема 3
РП (11) обладает следующим свойством:

$$\forall\, \theta \in \Theta_{\mathcal{P}} \;\; \theta \models E\bot\dagger, \text{ где } E = \{k_{BJ}, k_J, n_B^r \mid B \in Ag\}. \qquad (12)$$

Теорема 4

РП (11) обладает следующим свойством: $\forall\, R_B \in \mathcal{P}$, $\forall\, \theta \in \Theta_{\mathcal{P}}$, если $\theta \vdash at_{R_B} = 3$, то $\exists\ I_A \in \mathcal{P}$:

$$\theta \vdash \left\{ \begin{array}{l} at_{I_A} = 3 \\ a^r_A = B,\ a^i_B = A \\ n^i_A = n^i_B,\ n^r_A = n^r_B \\ k^i_A = k^r_B \end{array} \right\} \tag{13}$$

Теорема 5

РП (11) обладает следующим свойством: $\forall\, I_A \in \mathcal{P}, \forall\, \theta \in \Theta_{\mathcal{P}}$, если $\theta \vdash at_{I_A} = 2$, то $\exists\, R_B \in \mathcal{P}$:

$$\theta \vdash \left\{ \begin{array}{l} at_{R_B} = 2, \\ a^r_A = B, a^i_B = A, \\ n^i_A = n^i_B, n^r_A = n^r_B \end{array} \right\}. \tag{14}$$

Заключение

В настоящем докладе 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.