УДК 519.765
AMS MSC2020: 68Q45

Верификация криптографических протоколов[1]

Миронов А. М.

Университет Иннополис;
Лидирующий Исследовательский Центр

Аннотация. В работе излагается новая математическая модель криптографических протоколов, и приводится пример применения этой модели для верификации протоколов аутентификации.

Ключевые слова: криптографический протокол, процессная модель, верификация.

Введение

Понятие криптографического протокола

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

Для обеспечения свойств безопасности КП (таких например как конфиденциальность передаваемых данных) в КП могут использоваться криптографические преобразования (шифрование, электронная подпись, хэш-функции, и т. п.). Мы предполагаем, что криптографические преобразования, используемые в КП, являются идеальными, то есть удовлетворяют некоторым аксиомам, выражающим, например, невозможность извлечения открытых текстов из шифртекстов без знания соответствующих криптографических ключей.

[1]Исследование выполнено при финансовой поддержке Министерства цифрового развития, связи и массовых коммуникаций РФ и АО «Российская венчурная компания» (договор №004/20 от 20.03.2020, ИГК 0000000007119P190002)

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 [3], 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 [2], связанная с возможностью непредусмотренного нечестного поведения одного из участников этого КП и подрывающая безопасность этого КП. Особенность этой ошибки заключается в том, что данный КП является предельно простым распределенным алгоритмом, состоящим всего из трех действий, и при визуальном анализе этого КП отсутствие в нем ошибок не вызывало никаких сомнений. Ошибка была обнаружена лишь при помощи инструмента автоматизированной верификации КП.

Другой пример логической ошибки в КП [1]: 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.

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

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

- построение математических моделей анализируемых КП,
- описание свойств анализируемых КП в виде математических объектов, называемых спецификациями свойств этих КП, и
- построение формальных доказательств утверждений о том, что анализируемые КП удовлетворяют (или не удовлетворяют)

своим спецификациям, процедура построения таких доказательств называется верификацией анализируемых КП.

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

1. Последовательные и распределенные процессы

В этом параграфе мы излагаем понятия последовательного и распределенного процессов. Последовательный процесс является моделью участника КП, а распределенный процесс является моделью всего КП.

1.1. Вспомогательные понятия

1.1.1. Термы

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

Каждому элементу x множеств Con, Var и Fun сопоставлен некоторый тип $\tau(x) \in Types$, причем если $x \in Fun$, то $\tau(x)$ имеет вид $(\tau_1, \ldots, \tau_n) \to \tau$, где $\tau_1, \ldots, \tau_n, \tau \in Types$.

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

- x является термом типа $\tau(x)$ для любого $x \in Con \cup Var$,

- если $f \in Fun$, $e_1, \ldots, e_n$ — термы, и $\tau(f) = (\tau(e_1), \ldots, \tau(e_n)) \to \tau$, то запись $f(e_1, \ldots, e_n)$ является термом типа τ.

Будем использовать следующие обозначения:

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

1.1.2. Примеры типов

Будем считать, что $Types$ содержит следующие типы:

- тип $\mathbf{A}$, термы этого типа называются агентами,
- тип $\mathbf{C}$, термы этого типа называются каналами, они обозначают каналы связи, при помощи которых агенты взаимодействуют друг с другом путем передачи сообщений,
- тип $\mathbf{K}$, термы этого типа называются ключами, они обозначают криптографические ключи, которые агенты могут использовать для шифрования или дешифрования сообщений,
- тип $\mathbf{M}$, термы этого типа называются сообщениями, они обозначают сообщения, которые агенты могут пересылать друг другу во время своей работы,
- тип $\mathbf{N}$, термы этого типа называются нонсами, они обозначают переменные с уникальными значениями,
- тип $\mathbf{P}$, термы этого типа называются процессами.

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

Будем использовать следующие соглашения и обозначения:

- множество *Channels* содержит переменную, обозначаемую символом $\circ$, и называемую открытым каналом,

- тип $\mathbf{M}$ включает все другие типы из $Types$, то есть терм любого типа является также термом типа $\mathbf{M}$,
- для любых $n \geqslant 1, \tau \in Types$ множество $Types$ содержит тип τ_n, значения которого — кортежи длины n из значений типа τ.

1.1.3. Примеры функциональных символов

Будем предполагать, что Fun содержит следующие ФС.

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

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

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

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

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

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

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

  Термы вида $encrypt(k, e)$ и $decrypt(k, e)$ обозначают сообщения, получаемые шифрованием (и дешифрованием, соответственно) сообщения e на ключе k. Термы вида $encrypt(k, e)$ будут обозначаться записями $k(e)$, данные термы называются шифрованными сообщениями (ШС).

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

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

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

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

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

Будем использовать следующие обозначения: для любого $e \in Tm$

$$VarEncKeys(e) \quad = \{k \in Var_{\mathbf{K}} \mid \exists\, e' \in Tm : k(e') \subseteq e\}.$$

1.1.4. Выражения

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

Выражением называется запись одного из следующих видов:

- E, где $E \subseteq Tm$,
- $[P]$ и $[c]$, где $P \in Processes$, $c \in Channels$,
- $k^{-1}(E)$, где $k \in Keys$, и $E \in Expr$,
- $E \cap E'$, $E \cup E'$, $\neg E$, где $E, E' \in Expr$.

$Var(E) = \{x \in Var \mid x$ входит в $E\}$ для любого $E \in Expr$.

Выражения вида $k^{-1}([P])$ и $k^{-1}([c])$ обозначаются $k^{-1}[P]$ и $k^{-1}[c]$ соответственно. Выражения вида $\{e\}$, где $e \in Tm$, обозначаются без фигурных скобок. Ниже каждому выражению сопоставляется значение этого выражения в текущий момент времени, которое является множеством термов.

1.1.5. Формулы

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

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

ЭФ выражают свойства значений входящих в них выражений в текущий момент времени. ЭФ из первого пункта выражают свойства, соответствующие входящим в них теоретико-множественным символам. ЭФ из второго пункта выражают свойства, изложенные в пункте 1.2.4, ЭФ из третьего пункта выражают свойства текущего состояния последовательного процесса, подробнее см. в пункте 1.2.3.

Ф о р м у л о й называется произвольная совокупность ЭФ. Каждая формула $\varphi = \{\varphi_i \mid i \in I\}$ выражает утверждение, представляющее собой конъюнкцию утверждений, выражаемых ЭФ φ_i $(i \in I)$.

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

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

1.1.6. Связывания

С в я з ы в а н и е — это функция $\theta : Var \to Tm$. Будем говорить, что связывание θ связывает переменную $x \in Var$ с термом $\theta(x)$.

Будем использовать следующие обозначения:

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

$$x \mapsto \theta(x) \quad \text{или} \quad (\theta(x_1)/x_1, \ldots, \theta(x_n)/x_n), \tag{1}$$

  вторая запись в (1) используется, когда $\theta \in \Theta(\{x_1, \ldots, x_n\})$,
- для любых $\theta \in \Theta, e \in Tm$ запись e^θ обозначает терм, получаемый из e заменой для всех $x \in Var(e)$ каждого вхождения x в e на терм $\theta(x)$, терм e называется ш а б л о н о м терма e^θ относительно θ,
- для любых $\theta \in \Theta, E \subseteq Tm$ запись E^θ обозначает множество $\{e^\theta \mid e \in E\}$,

- для всех $\theta, \theta' \in \Theta$ запись $\theta\theta'$ обозначает связывание $x \mapsto (x^{\theta})^{\theta'}$.

Пусть $X \subseteq X' \subseteq Var$, $\theta \in \Theta(X)$, $\theta' \in \Theta(X')$. θ' называется продолжением θ, если $\theta(x) = \theta'(x)$ для любого $x \in X$.

1.2. Последовательные процессы

1.2.1. Действия

Действие — это запись одного из следующих видов:

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

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

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

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

1.2.2. Понятие последовательного процесса

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

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

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

Для каждого ПП $(P, A, X, \bar{X})$

- данный ПП может сокращенно обозначаться тем же символом P, что и соответствующий ему граф, множество вершин графа P также обозначается символом P,
- $Agent(P)$, $X(P)$, $\bar{X}(P)$ обозначают соответствующие компоненты P, $Var(P)$ обозначает множество всех переменных, входящих в P,
- $\tilde{X}(P)$ обозначает множество $X(P) \setminus \bar{X}(P)$ инициализированных нескрытых переменных процесса P,
- $\hat{X}(P)$ обозначает множество $Var(P) \setminus X(P)$ неинициализированных переменных процесса P.

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

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

1.2.3. Состояние последовательного процесса

Состояние ПП P — это пятерка

$$s = (at, \alpha, [P], \theta, \{[c] \mid c \in Channels\})$$

где

- $at \in P$ — вершина графа P в состоянии s,
- $\alpha \in \{init\} \sqcup Act$ — действие перед переходом в s,
- $[P] \subseteq Var$ — множество инициализированных переменных в s,
- $\theta \in \Theta([P])$ — связывание в s,

- для любого $c \in \mathit{Channels}$ $[c] \subseteq Tm$ — содержимое канала c в s.

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

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

1.2.4. Значения выражений и формул в состояниях последовательных процессов

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

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

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

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

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

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

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

$$\left.\begin{array}{l} \forall\, x \in E^s_{\mathbf{X}}, \forall\, y \in [P]_s \;\; x \notin y^s \\ \forall\, x \in E^s_{\mathbf{X}}, \forall\, c \in \mathit{Channels}, \\ \qquad\qquad \text{если } \exists\, e \in [c]_s : x \in e, \text{ то } c \in E^s \end{array}\right\} \qquad (2)$$

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

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

$$\left.\begin{array}{l} \forall\, x \in E^s_{\mathbf{X}},\, \forall\, y \in [P]_s \;\; x \perp_{\mathbf{K},E} y^s \\ \forall\, x \in E^s_{\mathbf{X}},\, \forall\, c \in Channels, \forall\, e \in [c]_s \;\; x \perp_{\mathbf{K},E} e \end{array}\right\} \qquad (3)$$

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

$$\begin{array}{l} \text{каждое вхождение } x \text{ в } e \text{ содержится} \\ \text{в подтерме } k(\ldots) \subseteq e, \text{ где } k \in E^s_{\mathbf{K}} \end{array} \qquad (4)$$

(3) можно интерпретировать как следующее утверждение: переменные из $E^s_{\mathbf{X}}$ входят в термы, доступные процессу P в состоянии s, а также в термы из содержимого произвольного канала, в «защищенном» виде, то есть входят в подтермы вида $k(\ldots)$, где $k \in E^s_{\mathbf{K}}$,

- $\varphi = (at_P = i)$, и $at_s = i$,
- $\varphi = \{\varphi_i \mid i \in I\}$ — совокупность ЭФ, и $s \models \varphi_i$ для любого $i \in I$.

1.2.5. Выполнение последовательного процесса

Выполнение ПП P можно понимать как обход вершин P, начиная с $Init(P)$, с выполнением действий, являющихся метками проходимых ребер. С каждым шагом выполнения ПП P связано некоторое состояние s ПП P, называемое текущим состоянием ПП P на этом шаге (на первом шаге текущим состоянием является 0_P). Если текущий шаг выполнения ПП P не является заключительным, то на этом шаге происходит замена текущего состояния s на состояние s', которое будет текущим состоянием на следующем шаге, для этого

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

- если α^{θ_s} содержит вхождение терма вида $shared\_key(\ldots)$, то $Agent(P)$ присутствует в этом вхождении,
- выполнено одно из условий:

$$\left.\begin{array}{ll} \text{(а)} & \alpha = c!e,\ c, e \in \langle P\rangle_s \\ \text{(б)} & \alpha = c?e,\ c \in \langle P\rangle_s, \\ & \quad VarEncKeys(e^s) \subseteq [P]_s, \\ & \quad \exists\, \theta \in \Theta(\, Var(e) \setminus [P]_s) : (e^\theta)^s \in [c]^s \\ \text{(в)} & \alpha = (e := e'),\ e' \in \langle P\rangle_s, \\ & \quad VarEncKeys(e^s) \subseteq [P]_s, \\ & \quad \exists\, \theta \in \Theta(\, Var(e) \setminus [P]_s) : e^\theta = e' \end{array}\right\} \qquad (5)$$

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

- если верно (а) в (5), то $[P]_{s'} = [P]_s$, $\theta_{s'} = \theta_s$, $[c^s]_{s'} = [c^s]_s \cup \{e^s\}$, $[c']_{s'} = [c']_s$ для любого $c' \in Channels \setminus \{c^s\}$,
- если верно (б) или (в) в (5), то $[P]_{s'} = [P]_s \cup Var(e)$, $\theta_{s'} = \theta\theta_s$, $[c']_{s'} = [c']_s$ для любого $c' \in Channels$, (будем говорить, что при переходе от s к s' каждая переменная $x \in Var(e) \setminus [P]_s$ инициализируется значением $x^{\theta_{s'}}$, которое становится доступным P),

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

Если имеет место первая (вторая) из указанных выше ситуаций, то будем говорить, что s' получается а к т и в н ы м (соответственно, п а с с и в н ы м) переходом из s. Запись $s \xrightarrow{P} s'$ ($s \to s'$) обозначает, что s' получается активным (соответственно, пассивным) переходом из s.

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

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

2) если $x \in \bar{X}(P)$, то это означает, что в начальный момент каждого выполнения $Exec$ ПП P данная переменная инициализирована уникальным значением, то есть значением, которое отличается от значений, сопоставленных другим инициализированным переменным при выполнении $Exec$, и от значений, сопоставленных инициализированным переменным при любом выполнении $Exec' \neq Exec$ любого ПП.

Интерпретация условий, описанных в (5), имеет следующий вид.

- Условие в (а) связано с выполнением посылки сообщения: имя c^s канала, в который посылается сообщение, должно быть доступно процессу P в состоянии s, и посылаемое сообщение e^s должно быть термом, компоненты которого также доступны процессу P в состоянии s.

- Условие в пункте (б) связано с выполнением приема сообщения:

  – имя c^s канала, из которого принимается сообщение, должно быть доступно процессу P в состоянии s,

  – все ШС в принимаемом сообщении, которые дешифруются во время приема этого сообщения, и зашифрованы не на разделяемом ключе, имеют вид $k(\ldots)$, где значение ключа k должно быть доступно процессу P в состоянии s, это свойство выражается во второй строке пункта (б) в (5),

  – терм e является шаблоном некоторого терма из $[c]^s$ относительно некоторого продолжения связывания θ_s, данное свойство выражается в последней строке пункта (б) в (5).

- Условие в пункте (с) связано с выполнением присваивания:

  – каждая компонента $(e')^s$ должна быть доступна P в s,

– смысл свойств во второй и третьей строках пункта (c) в (5) совпадает со смыслом соответствующих свойств в пункте (b): каждое ШС в $(e')^s$, которое должно быть дешифровано во время выполнения этого присваивания, должно иметь вид $k(\ldots)$, причем

  * либо k разделяемый ключ,

  * либо $k \in Var_{\mathbf{K}}$ и значение ключа k должно быть доступно P в состоянии s,

– терм e является шаблоном терма e' относительно некоторого связывания $\theta \in \Theta(\mathit{Var}(e) \setminus [P]_s)$.

1.2.6. Процесс противника

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

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

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

1.2.7. Переименование переменных

Переименование переменных (называемое также просто переименованием) — это инъективная функция $\eta : X \to X'$, где $X, X' \subseteq \mathit{Var}$. Для каждого переименования $\eta : X \to X'$, каждого $e \in \mathit{Tm}$ и каждого ПП P записи e^η и P^η обозначают терм или ПП соответственно, получаемые из e или P заменой для любого $x \in X$ каждого вхождения x на $\eta(x)$.

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

1.3. Распределенные процессы

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

1.3.1. Понятие распределенного процесса

Р а с п р е д е л е н н ы й п р о ц е с с (РП) — это семейство ПП:

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

(некоторые из которых могут совпадать). С каждым РП связана переменная типа **P**, называемая и м е н е м этого РП.

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

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

- $Var(\mathcal{P}) = \bigcup\limits_{P \in \mathcal{P}} Var(P)$, множества $X(\mathcal{P})$, $\bar{X}(\mathcal{P})$, $\tilde{X}(\mathcal{P})$, $\hat{X}(\mathcal{P})$ определяются аналогично,
- будем предполагать, что

$$\text{компоненты семейства } \{\bar{X}(P) \cup \hat{X}(P) \mid P \in \mathcal{P}\} \text{ дизъюнктны и не пересекаются с } \tilde{X}(\mathcal{P}) \tag{6}$$

  (если это не так, то заменим каждый из компонентов P семейства $\mathcal{P}$ на равный ему в том смысле, который указан в конце пункта 1.2.7, так, чтобы свойство (6) выполнялось),

- РП $\mathcal{P}$ может обозначаться записью
  - $\{P_1, \ldots, P_n\}$, если $I = \{1, \ldots, n\}$ (в случае $n = 1$ скобки могут быть опущены, то есть вместо $\{P_1\}$ пишется P_1), или
  - P^*, если I — множество натуральных чисел, и все ПП, входящие в $\mathcal{P}$, совпадают с P,

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

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

1.3.2. Понятие состояния распределенного процесса

Пусть задан РП $\mathcal{P}$.

С о с т о я н и е м РП $\mathcal{P}$ называется семейство $s = \{s_P \mid P \in \mathcal{P}\}$ состояний ПП, входящих в $\mathcal{P}$, такое, что для любого $c \in \mathit{Channels}$ все множества в семействе $\{[c]_{s_P} \mid P \in \mathcal{P}\}$ одинаковы (обозначаем их $[c]_s$).

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

- s называется н а ч а л ь н ы м состоянием РП $\mathcal{P}$, и обозначается $0_\mathcal{P}$, если $s_P = 0_P$ для любого $P \in \mathcal{P}$,

- $at_s = \{at_{s_P} \mid P \in \mathcal{P}\}$, $[\mathcal{P}]_s = \bigcup_{P \in \mathcal{P}} [P]_s$, $\langle \mathcal{P} \rangle_s = Tm([\mathcal{P}]_s)$,

- θ_s обозначает связывание из $\Theta([\mathcal{P}]_s)$ такое, что

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

существование такого связывания следует из (6).

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

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

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

1.3.3. Выполнение распределенного процесса

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

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

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

$$s_P \xrightarrow{P} s'_P, \quad \forall\, P' \in \mathcal{P} \setminus \{P\} \;\; s_{P'} \to s'_{P'}, \qquad \text{где } s = \{s_P \mid P \in \mathcal{P}\},\ s' = \{s'_P \mid P \in \mathcal{P}\}. \quad (7)$$

Свойство (7) обозначается записью $s \xrightarrow{\alpha_P} s'$, где $\alpha = \alpha_{s'_P}$.

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

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

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

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

1.4. Теорема для доказательства свойства соответствия

Теорема, излагаемая в этом параграфе, может использоваться для доказательства свойства соответствия протоколов

аутентификации, которое имеет следующий смысл: если один из участников протокола аутентификации после выполнения этого протокола пришел к выводу, что другой участник этого протокола является подлинным (то есть объявленные им свое имя и параметры совпадают с его реальными именем и параметрами), то это действительно так. Доказываемая ниже теорема применяется для обоснования того, что если РП $\mathcal{P}$ использует для взаимодействия только открытый канал $\circ$, и в некотором состоянии $s \in \Sigma_{\mathcal{P}}$ в этом канале содержится сообщение, содержащее подтерм вида $k(e)$, где ключ k недоступен в этом состоянии для некоторого ПП P, входящего в $\mathcal{P}$, то в некотором состоянии $s' <_\pi s$ другой ПП $P' \neq P$ из $\mathcal{P}$ послал в открытый канал $\circ$ сообщение, содержащее тот же самый подтерм $k(e)$.

Теорема 1. *Пусть заданы РП $\mathcal{P}$, такой, что $Var(\mathcal{P})_{\mathbf{C}} = \{\circ\}$, ПП $P \in \mathcal{P}$, множество $E \subseteq \langle\mathcal{P}\rangle_0$, не содержащее открытых ключей, и состояние $s \in \Sigma_{\mathcal{P}}$, причем $s \models E \perp_{\mathbf{K}} P$, и $[\circ]_s$ содержит терм с подтермом $k(e)$, где $k \in E_{\mathbf{K}}$.*

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

$$\dot{s} \xrightarrow{(!\dot{e})_{P'}} s', \quad \text{где } k(e) \subseteq \dot{e}^{\dot{s}}. \tag{8}$$

1.5. Схемы распределенных процессов

Пусть задан РП $\mathcal{P}$. Зависимости между действиями в $\mathcal{P}$ можно выразить в виде схемы РП $\mathcal{P}$, в которой каждый ПП $P \in \mathcal{P}$ представляется нитью, то есть вертикальной линией, на которой выделены точки, соответствующие вершинам из P. Пример схемы РП представлен диаграммой (10). В целях большей наглядности будем указывать в схемах РП горизонтальную черту над любым обозначением какой-либо переменной x, если она рассматривается как элемент множества $\bar{X}(P)$, то есть эта переменная обозначается $\bar{x}$.

2. Верификация протокола Yahalom

В этом параграфе рассматривается пример КП, который можно верифицировать на основе предлагаемого подхода путем использо-

вания теоремы 1.

2.1. Описание протокола Yahalom

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

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

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

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

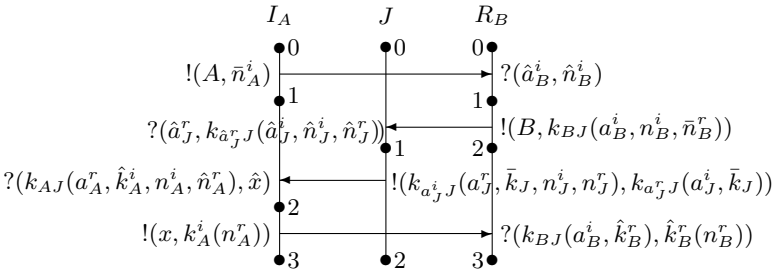
Пересылки в (9) имеют следующий смысл:

1) A посылает B запрос на аутентификацию и генерацию сеансового ключа k, запрос состоит из имени A и нонса n_A,
2) B посылает J запрос на генерацию сеансового ключа k, в запрос он включает свое имя, имя агента A, для связи с которым нужен этот ключ, полученный нонс n_A, и свой нонс n_B,
3) J генерирует сеансовый ключ k и посылает A пару сообщений, из первого сообщения A может извлечь сеансовый ключ k, а второе предназначено для того, чтобы A переслал его B,

4) A посылает B пару сообщений,

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

Формальное описание сеанса КП Yahalom изображается схемой



(10)

В этой схеме левая и правая нити соответствуют ПП I_A и R_B, описывающим поведение инициатора A и респондера B соответственно, средняя нить соответствует ПП, описывающему поведение посредника J, этот ПП обозначается тем же символом J. Смысл переменных в этих ПП усматривается из сопоставления действий в этих ПП с соответствующими действиями в (9). Верхний индекс i или r при какой-либо переменной означает, что она предположительно содержит информацию об инициаторе (i) или респондере (r) данного сеанса.

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

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

$$\mathcal{P} = \{\{I^*_A \mid A \in Ag\}, \{R^*_B \mid B \in Ag\}, J^*\}. \tag{11}$$

Ниже мы будем использовать следующее обозначение запись $s \models E \perp_{\mathbf{K}} e$ обозначает утверждение $\forall\, x \in E_{\mathbf{X}}\ \ x \perp_{\mathbf{K},E} e^s$.

2.2. Свойства протокола Yahalom

На основе предлагаемого подхода: могут быть верифицированы следующие свойства РП (11):

- секретность ключей и нонсов n_B^r:

$$\forall\, s \in \Sigma_{\mathcal{P}_\dagger} \quad s \models E \perp_{\mathbf{K}} P_\dagger, \quad \text{где } E = \{k_{BJ}, k_J, n_B^r \mid B \in Ag\} \quad (12)$$

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

$$s \models \{at_{I_A} = 3, a_A^r = B, a_B^i = A, n_A^i = n_B^i, n_A^r = n_B^r, k_A^i = k_B^r\}, \quad (13)$$

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

$$s \models \{at_{R_B} = 2, a_A^r = B, a_B^i = A, n_A^i = n_B^i, n_A^r = n_B^r\}. \quad (14)$$

Заключение

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

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

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

Список литературы

[1] *Cortier, V.* Formal Models and Techniques for Analyzing Security Protocols: A Tutorial / V. Cortier, S. Kremer // Foundations and Trends in Programming Languages. — 2014. — Vol. 1, №3. — P. 151–267.

[2] *Lowe, G.* An attack on the Needham-Schroeder public key authentication protocol // Information Processing Letters. — 1995. — Vol. 56, №3. — P. 131–133.

[3] *Needham, R.* Using encryption for authentication in large networks of computers / R. Needham, M. Schroeder // Communications of the ACM. — 1978. — Vol. 21, №12. — P. 993–999.

Библиографическая ссылка

Миронов, А. М. Верификация криптографических протоколов // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 213–234.
https://doi.org/10.26456/mfcsics-21-31

Сведения об авторах

Андрей Михайлович Миронов
Университет Иннополис;
Лидирующий Исследовательский Центр. Доцент

Россия, 420500, Иннополис, ул. Университетская, 1
E-mail: amironov66@gmail.com