УДК 004.056.5, 004.94 DOI 10.17223/2226308X/14/27

О ПРИЁМАХ ПО ДОРАБОТКЕ СОГЛАСОВАННОГО ОПИСАНИЯ МРОСЛ ДП-МОДЕЛИ ДЛЯ ОС И СУБД С ЦЕЛЬЮ ЕГО ВЕРИФИКАЦИИ ИНСТРУМЕНТАМИ Rodin И ProB

П. Н. Девянин, М. А. Леонова

Рассматриваются приемы согласованного описания мандатной сущностно-ролевой ДП-модели управления доступом и информационными потоками в операционных системах семейства Linux (МРОСЛ ДП-модели) в математической и формализованной нотациях с целью обеспечения возможностей для, во-первых, её совместной верификации дедуктивным методом и методом проверки моделей (*model checking*) с применением инструментов Rodin и ProB соответственно, во-вторых, моделирования на формализованном языке метода Event-B взаимодействующих между собой систем с собственными развитыми механизмами управлениями доступом, таких, как ОС и СУБД, что необходимо для соответствия описанию модели в математической нотации.

Ключевые слова: *формальная модель управления доступом, верификация, Event-B, требования доверия, Astra Linux Special Edition.*

Введение

В средствах защиты информации (СЗИ), таких, как ОС или СУБД, механизм управления доступом выполняет одну из основных функций по обеспечению их безопасности. Для достижения доверия к корректности этого механизма, создания условий для научного обоснования выполнения им заданных для СЗИ требований безопасности уже многие десятилетия разрабатываются формальные модели управления доступом [1–3].

Изначально эти модели формировались на математическом языке (в математической нотации), вместе с этим по мере увеличения объёмов описания формальных моделей, в связи с необходимостью устранения в них ошибок, для описания формальных моделей стали применяться формализованные (машиночитаемые) языки (формализованная нотация), например язык формального метода Event-B [4]. При этом для автоматизированной проверки корректности и верификации описания моделей в формализованной нотации начали использоваться соответствующие инструменты, например инструмент дедуктивной верификации Rodin [5].

В настоящее время требования по разработке формальных моделей управления доступом и их верификации стали частью актуальных нормативных документов, определяющих требования доверия к сертифицированным СЗИ. Так, согласно утверждённым Приказом ФСТЭК России № 76 от 02.06.2020 во второй редакции «Требованиям по безопасности информации, устанавливающим уровни доверия к средствам технической защиты информации и средствам обеспечения безопасности информационных технологий» [6], механизм управления доступом СЗИ должен разрабатываться на основе формальной модели управления доступом (начиная с четвёртого уровня доверия), которая должна быть верифицирована с применением инструментальных средств (начиная с третьего уровня доверия). Кроме того, для стандартизации применяемых для этого методов и технологий недавно утверждены два национальных стандарта ГОСТ Р 59453.1-2021 и ГОСТ Р 59453.2-2021 [7, 8].

Поскольку разрабатываемая ООО «РусБИТех-Астра» (ГК Astra Linux) операционная система специального назначения (ОССН) Astra Linux Special Edition [9, 10]

в системе сертификации ФСТЭК России сертифицирована по высшему первому классу защиты (первому уровню доверия), её подсистема безопасности PARSEC создана на основе мандатной сущностно-ролевой ДП-модели управления доступом и информационными потоками в ОС семейства Linux (МРОСЛ ДП-модели) [3]. Её описание в математической нотации составляет более 500 страниц текста. Она имеет иерархическое представление, состоящее из восьми уровней — четырёх уровней для моделирования управления доступом непосредственно в ОССН и четырёх аналогичных уровней для штатной для ОССН СУБД PostgreSQL.

Отмеченные объём и сложность описания и, как следствие, проверки корректности МРОСЛ ДП-модели в математической нотации стали стимулом для её описания в формализованной нотации на языке метода Event-B и автоматизированной дедуктивной верификация с применением инструмента Rodin [11]. В этой нотации текущее представление модели также имеет значительный объём (более 15 тысяч строк кода). Кроме того, для повышения качества модели авторами освоена технология верификации формальной модели с применением метода проверки моделей (*model checking*), реализованном в инструменте ProB [12]. Всё перечисленное требует поиска приёмов и технологий согласованного для ОС и СУБД описания МРОСЛ ДП-модели в математической и формализованной нотациях, обеспечения условий для её автоматизированной верификации дедуктивным методом и методом проверки моделей. Анализу возникающих здесь проблем и выработанным авторами новым техническим приёмам для их решения посвящается настоящая работа.

1. Согласованное описание модели для ОС и СУБД в математической и формализованной нотациях

В формализованной нотации МРОСЛ ДП-модель разрабатывается в виде последовательности связанных между собой спецификаций, соответствующих уровням в её математической нотации, с использованием техники пошагового уточнения (*refinement*) [13] на языке формального метода Event-B, что позволяет повысить качество модели и устранить возможные ошибки. В каждой спецификации элементам, задающим в рамках соответствующего уровня состояния моделируемой ОССН, в модели в формализованной нотации ставятся в соответствие переменные Event-B, а правилам перехода из состояний в состояния — события Event-B. Инварианты на переменные описывают свойства внутренней согласованности элементов соответствующей спецификации модели, в том числе свойства безопасности. При этом доказательство того, что любой переход из состояния в состояние, заданный событиями Event-B, сохраняет все инварианты состояния, а значит, корректности описания модели и выполнения в её рамках условий безопасности, осуществляется с применением инструмента дедуктивной верификации Rodin.

Накопленный опыт формирования МРОСЛ ДП-модели в формализованной нотации, проведённые авторами научные исследования по данному направлению выявили ряд недостатков в используемых для этого технологиях и приёмах, такие, как часто независимое друг от друга описание элементов модели для различных видов управления доступом, а также многократное дублирование одинаковых условий их выполнения, не соответствующее реализации проверок этих условий непосредственно в программном коде ОССН; сложность добавления уровней уточнений, моделирующих взаимодействующие с ОССН системы, ввиду того, что в изначально использованном подходе по формированию структуры элементов модели [11] не предусматривалось логического разделения элементов, предназначенных для моделирования различных,

но взаимодействующих между собой механизмов управления доступом, например, реализуемых PARSEC в ОССН и СУБД PostgreSQL.

Для устранения данных недостатков предложен ряд приёмов [14]: логическое объединение проверок условий, заданных для мандатных управления доступом, контроля целостности и ролевого управления доступом путём применения тотальных функций формализованного языка метода Event-B, а также использование при построении структуры элементов модели подтипов формализованного языка метода Event-B, которые позволяют раздельно описывать такие существенные для моделирования управления доступом элементы, как субъекты (процессы), сущности (файлы, каталоги, базы данных), учётные записи пользователей, роли, информационные потоки.

В процессе добавления уровней уточнений были выявлены трудности, связанные с применяемой Rodin техникой *refinement*, а именно строго последовательное (линейное) уточнение уровней. Это не соответствует иерархическому представлению МРОСЛ ДП-модели в математической нотации, в котором каждый из уровней моделируемого механизма управления доступом взаимодействующей с ОССН системы уточняет предыдущий для неё уровень, а также соответствующий уровень аналогичного механизма самой ОССН (например, в модели уровень мандатного контроля целостности СУБД основан на уровнях ролевого управления доступом СУБД и мандатного управления доступом ОССН). Ещё одним ограничением, накладываемым логикой пошагового уточнения, является отсутствие возможности изменения значений определяемых инвариантами функций на любых уровнях спецификаций, кроме того, на котором они задаются, хотя это необходимо для корректного моделирования взаимодействия систем (например, возникновения информационных потоков между элементами ОССН и СУБД).

В результате потребовалось найти такой приём по уточнению уровней модели в формализованной нотации, который, с одной стороны, был бы логически согласован с порядком уточнения уровней иерархического представления модели в математической нотации, а с другой стороны, использовал бы имеющуюся в Rodin технику *refinement* для сохранения возможности дедуктивной верификации модели. Этим решением стало последовательное уточнение уровней спецификаций модели для ОССН уровнями взаимодействующей с ней системы, но с переопределением на них необходимых для согласованности модели в математической и формализованной нотациях функций и событий. Например, переопределение на втором уровне модели для СУБД заданных на аналогичном уровне ОССН функций информационных потоков по памяти от субъектов к сущностям $SEMFlows$, от субъектов к субъектам $SSMFlows$, де-факто события создания информационного потока по памяти от субъекта к сущности при наличии субъекта-посредника $find\_entity\_m$, и задания новых функций $dbSEMFlows$, $dbSSFlows$, а также соответствующих де-факто событий для ОС os_find_entity и для СУБД db_find_entity (листинг 1).

```
os_find_entity
ANY
x, y, z, flow

WHERE
grd1: x ∈ Subjects
grd2: y ∈ Subjects
grd3: z ∈ Entities
grd4: flow ∈ ℙ1({1,2})
//‘‘1’’ - информационный поток по памяти, ‘‘2’’ - информационный поток по времени
```

```
grd5: 1 ∈ flow ⇒ z ∉ EHole
grd6: 1 ∈ flow ⇒ y ↦ 1 ∈ dbSSFlows(x) ∧ z ↦ 1 ∈ dbSEFlows(y)

THEN
act1: dbSEFlows(x) := dbSEFlows(x) ∪ ({z} × flow)

db_find_entity
ANY
x, y, z, flow

WHERE
grd1: x ∈ Subjects
grd2: y ∈ Subjects
grd3: z ∈ Entities
grd4: flow ∈ ℙ1({1,2})
grd5: 1 ∈ flow ⇒ z ∉ DBEHole
grd6: 1 ∈ flow ⇒ y ↦ 1 ∈ dbSSFlows(x) ∧ z ↦ 1 ∈ S_DBEFlows(y)

THEN
act1: S_DBEFlows(x) := S_DBEFlows(x) ∪ ({z} × flow)
```

Листинг 1. Общие для ОС и СУБД де-факто события

Достоинством данного приёма является возможность сохранять уровни спецификаций для моделируемого механизма управления доступом ОССН отдельно от соответствующих уровней для других систем. При этом он обладает и явным недостатком, заключающимся в необходимости повторения уже проведённых доказательств и увеличении объёма кода описания модели в формализованной нотации.

2. Совместная верификация инструментами Rodin и ProB

Для повышения качества верификации МРОСЛ ДП-модели в формализованной нотации, расширения спектра применяемых для этого методов и инструментов, моделирования и в перспективе автоматизированного тестирования на соответствие этой модели её реализации непосредственно в программном коде и настройках конфигурации механизма управления доступом ОССН осуществляется верификация модели с использованием инструмента проверки моделей ProB [12].

Как и для инструмента Rodin, здесь также выявлены трудности непосредственного применения ProB, связанные с большим объёмом описания модели в её формализованной нотации и использованием ряда способов представления модели на формализованной языке метода Event-B, не вызывавших затруднений при её дедуктивной верификации инструментом Rodin, но оказавшихся непригодными для применения ProB. Так, его работа часто завершалась с ошибкой вида `timeout`, вызванной превышением допустимого интервала времени, установленного для выполнения ProB переборных алгоритмов (ошибка такого вида является следствием проявления «комбинаторного взрыва»).

Это, в свою очередь, потребовало проведения исследований и разработки соответствующих приёмов [15], позволяющих осуществлять согласованную верификацию описания всех восьми уровней МРОСЛ ДП-модели в формализованной нотации инструментами проверки моделей ProB и дедуктивной верификации Rodin.

Примером одного из таких приёмов является использование нескольких несущих множеств (глобальных типов, над которыми не допускаются операции объединения, пересечения, разности, дополнения, какие-либо дополнительные ограничения на эти

множества могут быть заданы только аксиомами), но с сохранением подхода по применению подтипов [14]. С комбинаторной точки зрения полезно рационально определять типы, уходя от использования одного глобального типа для всех элементов модели, но и сохраняя преимущества применения подтипов там, где это необходимо (например, разделяя глобальные типы для сущностей и ролей и при этом используя отдельные подтипы глобального типа ролей для обычных, административных и запрещающих ролей).

Примером способа представления модели на формализованном языке метода Event-B, не вызывавшего затруднения при её дедуктивной верификации инструментом Rodin, но оказавшегося непригодными для применения ProB, является также описанный в [11] способ, позволяющий применять при доказательствах математическую индукцию, что в Rodin сделать непосредственно невозможно. Данный способ заключается во внесении в контекст первого уровня (ролевого управления доступом) формализованной нотации хорошо известной в математике аксиомы (листинг 2), где $\mathbb{N}$ — множество натуральных чисел, включая нуль. При этом в событиях, требующих доказательства инварианта с применением индукции, добавляются специальный параметр-функция и ряд охранных условий, проверяющих условия истинности инварианта, используя описанную аксиому. Например, в событии передачи административных прав доступа *grant_admin_rights* параметром-функцией *depth* задаются потомки роли *role* (роли, стоящие ниже неё в иерархии) и охранными условиями *grd*16–*grd*19 проверяются отношения уровней целостности данных потомков (листинг 3), т. е. роль, находящаяся выше в иерархии, должна иметь уровень целостности не ниже уровней целостности её потомков.

```
InductionAxiom: ∀s · s ⊆ ℕ ∧ 0 ∈ s ∧ (∀n · n ∈ s ⇒ n + 1 ∈ s) ⇒ ℕ ⊆ s
```

Листинг 2. Задание аксиомы индукции

```
invariant RParents4: ∀r, p · r ∈ Roles ∧ p ∈ RParents(r) ⇒ RoleInt(r) ⊆ RoleInt(p)

grant_admin_rights
ANY
depth

WHERE
grd16: depth ∈ ℕ → ℙ(dom(admRights))
grd17: ∀r · r ∈ dom(admRights) ⇒ (∃i · i ∈ ℕ ∧ r ∈ depth(i))
grd18: depth(0) = {role}
grd19: ∀i · i ∈ ℕ ⇒ (∀r · r ∈ depth(i + 1) ⇒ (∃p · p ∈ depth(i) ∧ p ∈ RParents(r)))
theorem grd20: ∀i · i ∈ ℕ ∧ (∀r · r ∈ depth(i) ⇒ RoleInt(r) ⊆ RoleInt(admRole)) ⇒
(∀r · r ∈ depth(i + 1) ⇒ RoleInt(r) ⊆ RoleInt(admRole))
theorem grd21: ∀i · i ∈ ℕ ⇒ (∀r · r ∈ depth(i) ⇒ RoleInt(r) ⊆ RoleInt(admRole))
```

Листинг 3. Задание потомков роли в иерархии с использованием индукции

При попытке задать контекст первого уровня и подобрать параметры для событий, включающих в себя данную структуру, ProB завершает работу с ошибкой вида `timeout`. Приёмом, позволяющим, с одной стороны, сохранить полноту дедуктивной верификации модели инструментом Rodin, а с другой — иметь возможность её верификации по методу проверки моделей с применением ProB, является задание параметра-функции в виде тотальной функции, один из инвариантов-истинности

которой должен описывать шаг индукции. Далее при доказательстве сохранения в соответствующих событиях основанных на индукции инвариантов использовать такую тотальную функцию. Например, для задания потомков роли без явного использования индукции применяются инварианты $DescendantsRType$, $DescendantsR1$, $DescendantsR2$, $NoCyclesForRoles$, задающие функцию потомков роли, и инвариант $DescendantsR6$, описывающий соответствующее шагу индукции условие, при котором уровень целостности роли должен быть не ниже уровня целостности каждого ее потомка (листинг 4).

```
   DescendantsRType: DescendantsR ∈ Roles → ℙ(Roles)
   DescendantsR1: ∀r,d · r ∈ Roles ∧ d ∈ Roles ⇒ (d ∈ DescendantsR(r) ⇔
d = r ∨ (d ≠ r ∧ (∃k · k ∈ KidsR(r) ∧ d ∈ DescendantsR(k))))
   DescendantsR2: ∀r,d · r ∈ Roles ∧ d ∈ Roles ⇒ (d ∈ DescendantsR(r) ⇔
DescendantsR(d) ⊆ DescendantsR(r))
   NoCyclesForRoles: ∀r,d · r ∈ Roles ∧ d ∈ Roles ∧ d ∈ DescendantsR(r) ∧
d ≠ r ⇒ r ∉ DescendantsR(d)
   DescendantsR6: ∀r, p · r ∈ Roles ∧ p ∈ Roles ∧ r ∈ DescendantsR(p) ⇒
RoleInt(r) ⊆ RoleInt(p)
```

Листинг 4. Задание потомков роли в иерархии с использованием тотальной функции

Выводы

В результате проведённых исследований расширен спектр технических приёмов, позволяющих осуществлять согласованное описание для ОС и СУБД уровней иерархического представления МРОСЛ ДП-модели в математической и формализованной нотациях на языке метода Event-B. Эти приёмы базируются на выражении в последовательном уточнении уровней модели на основе техники *refinement* свойств исходного иерархического представления модели и на применении тотальных функций вместо непосредственного использования аксиомы математической индукции. В итоге эти приёмы создали дополнительные условия для успешной верификации модели в формализованной нотации по дедуктивному методу инструментом Rodin и методу проверки моделей инструментом ProB. Предложенные приёмы могут быть полезны при разработке других формальных моделей управления доступом и их верификации с применением соответствующих инструментов.

ЛИТЕРАТУРА

1. *Bell D. E. and LaPadula L.J.* Secure Computer Systems: Unified Exposition and Multics Interpretation. Bedford, Mass.: MITRE Corp., 1976.
2. *Biba K. J.* Integrity Considerations for Secure Computer Systems. Bedford, Mass.: MITRE Corp., 1975.
3. *Девянин П. Н.* Модели безопасности компьютерных систем. Управление доступом и информационными потоками: учеб. пособие для вузов. 3-е изд., перераб. и доп. М.: Горячая линия — Телеком, 2020. 352 с.
4. *Abrial J.-R.* Modeling in Event-B: System and Software Engineering. Cambridge University Press, 2010.
5. *Abrial J.-R., Butler M., Hallerstede S., et al.* Rodin: An open toolset for modelling and reasoning in Event-B // Intern. J. Software Tools for Technology Transfer. 2010. V. 12. No. 6. P. 447–466.
6. Выписка из Требований по безопасности информации, утвержденных приказом ФСТЭК России № 76 от 02.06.2020. https://fstec.ru/tekhnicheskaya-zashchita-informatsii/dokumenty-po-sertifikatsii/120-normativnye-dokumenty/2126-

vypiska-iz-trebovanij-po-bezopasnosti-informatsii-utverzhdennykh-prikazom-fstek-rossii-ot-2-iyunya-2020-g-n-76.

7. ГОСТ Р 59453.1-2021 «Защита информации. Формальная модель управления доступом. Ч. 1. Общие положения». М.: Стандартинформ, 2021. 35 с.
8. ГОСТ Р 59453.2-2021 «Защита информации. Формальная модель управления доступом. Ч. 2. Рекомендации по верификация формальной модели управления доступом». М.: Стандартинформ, 2021. 23 с.
9. Операционная система специального назначения Astra Linux Special Edition. https://astralinux.ru/products/astra-linux-special-edition/
10. *Буренин П. В., Девянин П. Н., Лебеденко Е. В. и др.* Безопасность операционной системы специального назначения Astra Linux Special Edition: учеб. пособие для вузов / под ред. П. Н. Девянина. 3-е изд., перераб. и доп. М.: Горячая линия — Телеком, 2019. 404 с.
11. *Девянин П. Н., Ефремов Д. В., Кулямин В. В. и др.* Моделирование и верификация политик безопасности управления доступом в операционных системах. М.: Горячая линия — Телеком, 2019. 214 с.
12. *Leuschel M. and Butler M.* ProB: an automated analysis toolset for the B method // Int. J. Softw. Tools Technol. Transf. 2008. No. 10(2). P. 185–203.
13. *Abrial J.-R. and Hallerstede S.* Refinement, decomposition, and instantiation of discrete models: Application to Event-B // Fundamenta Informaticae. 2007. V. 77. Iss. 1–2. P. 1–28.
14. *Девянин П. Н., Леонова М. А.* Применение подтипов и тотальных функций формального метода Event-B для описания и верификации МРОСЛ ДП-модели // Программная инженерия. 2020. Т. 11. № 4. С. 230–241.
15. *Девянин П. Н., Леонова М. А.* Приёмы по доработке описания модели управления доступом ОССН Astra Linux Special Edition на формализованном языке метода Event-B для обеспечения ее автоматизированной верификации с применением инструментов Rodin и ProB // Прикладная дискретная математика. 2021. № 52. С. 83–96.

УДК 004.75 DOI 10.17223/2226308X/14/28

МЕТОД ОБЕСПЕЧЕНИЯ КОНФИДЕНЦИАЛЬНОСТИ ДАННЫХ НА ОСНОВЕ zk-SNARK[1]

Д. О. Кондырев

Представлен метод обеспечения конфиденциальности данных с возможностью проверки корректности на основе протокола доказательства с нулевым разглашением zk-SNARK. Разработанный метод позволяет создавать алгоритмы на основе zk-SNARK в смарт-контрактах Ethereum, используя высокоуровневые базовые криптографические схемы.

Ключевые слова: *распределённые системы, блокчейн, доказательство с нулевым разглашением, zk-SNARK, платформа Ethereum.*

Среди технических проблем, препятствующих внедрению технологии распределённых реестров, масштабируемость и конфиденциальность являются особенно существенными. В настоящий момент ведутся активные исследования, направленные на поиск решения проблемы конфиденциальности.

[1]Работа выполнена при поддержке Математического центра в Академгородке, соглашение с Министерством науки и высшего образования Российской Федерации № 075-15-2019-1613, и лаборатории криптографии JetBrains Research.