УДК 519.681.2

А.А. Лепендин, А.В. Уберт

Метод верификации моделей в приложении к анализу протоколов аутентификации

A.A. Lependin, A.V. Ubert

The Model Checking Method for the Authentication Protocols Analysis

Показана возможность применения метода model checking для верификации криптографических протоколов. На примере протокола Нидхема-Шрёдера продемонстрирован подход к моделированию взаимодействующих сторон как параллельно исполняемых процессов, обменивающихся сообщениями. Приведен пример формализации и проверки свойств протокола с помощью формул линейной темпоральной логики. Найдена классическая уязвимость протокола Нидхема-Шрёдера – атака Лоу.

Ключевые слова: криптографический протокол, аутентификация, метод model checking, линейная темпоральная логика, автоматическая верификация.

The research shows some capabilities to use model checking method for cryptographic protocol verification. An approach to modeling interacting agents as an asynchronous parallel processes was illustrated on an example of Needham-Schröeder protocol. An example of formalization and properties verification by means of using linear temporal logic was given. The well-known vulnerability of Needham-Schröeder protocol – Lowe attack – was found.

Key words: cryptographic protocol, authentication, model checking, linear temporal logic, automatic verification.

Развитие и совершенствование современных криптографических протоколов привело к тому, что их структура представляет собой относительно сложный набор правил взаимодействий общающихся между собой сторон. Имеется большая вариативность в выборе пути взаимодействия между легитимными участниками протокола. В случае присутствия злоумышленника число возможных сценариев, в некоторых случаях приводящих к осуществлению успешных атак, становится очень большим. Это обусловливает необходимость полной или частичной автоматизации процесса перебора и анализа различных сценариев взаимодействия сторон, что может позволить эффективнее обнаруживать потенциальные уязвимости как существующих, так и только разрабатываемых криптографических протоколов.

Существует достаточно широкий спектр методов, предназначенных для анализа компьютерных протоколов. Так как можно провести прямую аналогию между взаимодействиями при работе протокола и асинхронными параллельными вычислениями с участием нескольких сторон, то многие методы анализа являются продолжением и развитием подходов, разработанных для оценки эффективности и надежности параллельных вычислительных систем. Развивается индуктивный подход к анализу протоколов [1, с. 31], основанный на применении формальных логик и систем аксиом для полуавтоматического доказательства теорем о свойствах рассматриваемых криптографических систем. Существуют простые схемы анализа, основанные на расширении обычной математической логики понятием доверия сторон – BAN-логики [2, с. 2]. Непосредственно к данным подходам примыкает и метод, который применялся в данной работе – метод верификации моделей (принятый в русскоязычной литературе перевод термина model checking – см.: [3, с. 64]) и его логическое развитие в виде метода вероятностной верификации [4, с. 416].

Метод model checking основан на построении формальной модели системы (в нашем случае – криптографического протокола) с применением темпоральных логик того или иного вида [3, с. 56]. Их применение позволяет строить высказывания относительно развития процесса взаимодействия сторон в протоколе и выполнения или невыполнения определенных свойств как в течение всего сеанса взаимодействия, так и после определенных его этапов. В качестве основных обычно выбирают либо логику линейного времени (linear temporal logic – LTL), позволяющую говорить в первую очередь о глобальных свойствах протокола, либо логику деревьев вычислений (computational tree logic – CTL), где акцент делается на процессе вычисления или взаимодействия и его свойствах на отдельных этапах. Второй особенностью данного метода является его высокая эффективность при анализе дискретных систем с огромным числом возможных состояний (до $\sim 10^{120}$ согласно [3, с. 30]), что позво-

ляет применять этот метод для сложных протоколов безопасности со многими одновременно взаимодействующими сторонами. Существующее на нынешний момент разнообразие программных сред, поддерживающих работу с той или иной логикой, обеспечивает богатые возможности для анализа различных систем, в том числе и криптографических протоколов.

Цель данной работы – изучение возможностей проверки свойств криптографических протоколов с помощью метода верификации моделей на примере классического протокола Нидхема-Шредера с открытым ключом. В качестве программной среды для моделирования использовался Spin – свободно распространяемый пакет программ для формальной верификации систем [4, с. 197], имеющий несколько режимов работы, позволяющих проводить как автоматическую, так и ручную симуляцию, а также верификацию заданных высказываний, выраженных в LTL-логике относительно свойств криптографических протоколов.

Моделирование протокола основывалось на подходе, развитом в [4, с. 279], где также проводился анализ протокола Нидхема-Шредера. Отличием данной работы явилось то, что нами была создана полная, а не частичная модель всех этапов работы протокола, включая процесс взаимодействия легальных участников с третьей доверенной стороной. Схема взаимодействия сторон представлена на рисунке 1. Этапы 1, 2, 4 и 5 отвечают за получение Алисой и Бобом открытого ключа от Трента (третьей стороны), а этапы 3, 6 и 7 – за взаимную аутентификацию сторон – Алисы и Боба (разделение основано на [5, с. 81]). Каждому из участников протокола (взаимодействующие легальные пользователи Алиса и Боб, третья доверенная сторона Трент и Злоумышленник) ставился в соответствие модельный процесс. Все процессы выполнялись параллельно.

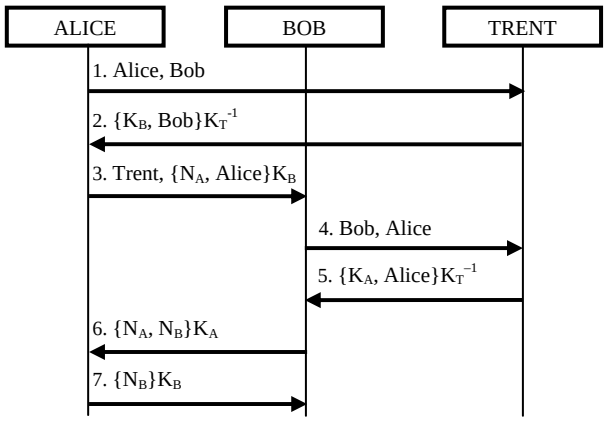

Рис. 1. Схема асимметричного протокола Нидхема-Шредера. K_A, K_B – открытые ключи Алисы и Боба; K_T^{-1} – закрытый ключ Трента; N_A, N_B – одноразовые случайные числа

Модель каждого легального пользователя включала в себя четыре основных блока. Первый блок – выбор партнера (только у инициатора обмена сообщениями – Алисы), второй блок – запрос публичного ключа партнера у Трента, третий – конструирование сообщения и отправка, последний – проверка сообщения. После обмена «секретами» каждый из процессов завершал работу. Было зарезервировано четыре канала связи: канал 1 – для взаимодействия легальных пользователей с третьей доверенной стороной, 2 и 3 предназначались для отправки сообщений от легитимных пользователей Злоумышленнику, 4 – для получения от него сообщений. Подобное разделение одновременно позволило как упростить структуру передаваемых сообщений, так и выделить отдельный «защищенный» канал связи с третьей доверенной стороной.

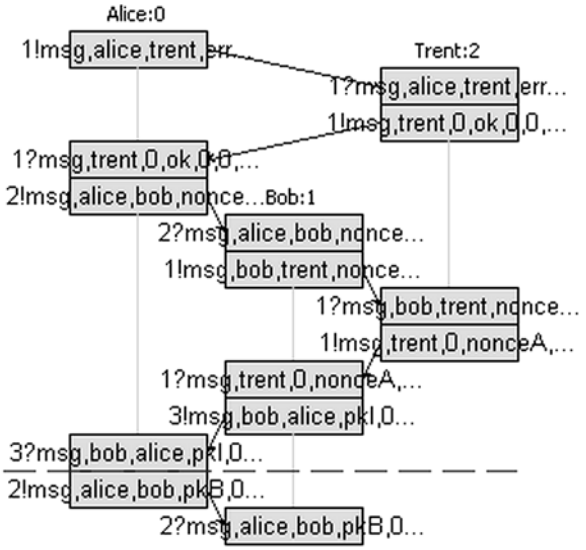

Рис. 2. Симуляция корректного сеанса протокола Нидхема-Шредера

На рисунке 2 приведен пример симуляции протокола без вмешательства Злоумышленника. Алиса запрашивает публичный ключ Боба у Трента и отправляет Бобу свой секрет, зашифрованный его ключом, затем Боб, получив сообщение, запрашивает ключ Алисы, отправляет ей свой секрет, и Алиса подтверждает установку доверительного соединения.

Далее вводился процесс, моделирующий Злоумышленника. К возможностям Злоумышленника относились прослушивание сообщений, сохранение перехваченных сообщений, если сообщения предназначались ему, то он мог расшифровать скрытую часть, иначе – хранил в том виде, в котором ее получил, генерирование сообщений на основе имеющейся информации, отправка своих сообщений в сеть. Атакой считалась ситуация, когда «сторона B считает своим партнером сторону A, сторона A доверяет своему партнеру, злоумышленник знает обе

части их секрета», что в виде формулы линейной темпоральной логики LTL выглядело следующим образом:

$$F(agentB\_finished \wedge bobtrustsA \wedge \wedge\ intrknowNA \wedge intrknowNB),$$

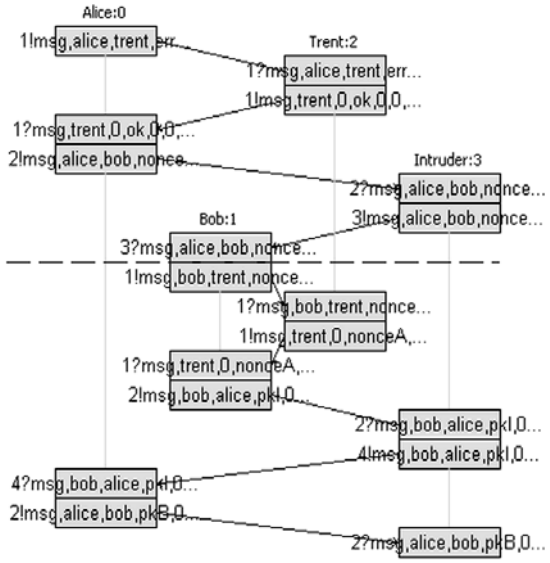

Рис. 3. Симуляция найденной атаки на алгоритм Нидхема-Шрёдера

где agentB_finished соответствует высказыванию «Боб корректно завершил сеанс взаимодействия», bobtrustsA – «Боб доверяет Алисе»; intrknowNA – «Злоумышленник знает секрет Алисы»; intrknowNB – «Злоумышленник знает секрет Боба». Квантор «F» означает, что данное высказывание будет справедливо в некий будущий момент.

В результате работы процедуры верификации был найден путь вычисления (взаимодействия сторон), при котором данная формула оказывалась истинной. На рисунке 3 показана соответствующая последовательность отсылаемых сообщений, приводящая к атаке. Сопоставление с литературными данными о возможных атаках на протокол Нидхема-Шредера [5, с. 83] показало, что была получена классическая атака Лоу.

Таким образом, развиваемый подход к моделированию криптографических протоколов как совокупностей параллельных процессов, взаимодействующих путем пересылки сообщений, в сочетании с возможностями метода model checking для полуавтоматического анализа уязвимостей позволил эффективно найти одну из наиболее сложных атак на протокол Нидхема-Шредера. Данный подход представляется крайне перспективным и может применяться к широкому кругу задач, связанных с проверкой защищенных сетевых протоколов.

Библиографический список

1. Bella G. Formal Correctness of Security Protocols. – Springer, 2007.

2. Mullender S.J. BAN Logic. A Logic of Authentication [Электронный ресурс]. – URL: http://www.home.cs.utwente. nl/~sape/sse/ban.pdf.

3. Кларк Э.М., Грамберг мл. О., Пелед Д. Верификация моделей программ. – М., 2002.

4. Карпов Ю. Г. Model Checking. Верификация параллельных и распределенных программных систем. – СПб., 2009.

5. Мао В. Современная криптография. Теория и практика. – М., 2005.