3. Богушевський В.С., Сухенко В.Ю. Керування режимом дуття конвертерної плавки // Наукові вісті НТУУ „КПІ”. – 2009. – № 1. – С. 58 – 64.

4. Богушевський В.С., Сергеєва К.О. Контроль температурного режиму конвертерної плавки // Наукові вісті НТУУ „КПІ”. – 2009. – № 6. – С. 75 – 80.

5. Богушевский В.С., Сухенко В.Ю., Сергеева Е.А. Управление доводкой конвертерной плавки // Металл и литье Украины. – 2010. – № 3. – С. 14 – 17.

УДК 001.57:004.942

ФОРМАЛИЗАЦИЯ СВОЙСТВ БЕЗОПАСНОСТИ КРИПТОГРАФИЧЕСКИХ ПРОТОКОЛОВ С ПОМОЩЬЮ ЛОГИКИ ЛИНЕЙНОГО ВРЕМЕНИ

В.А. Борхаленко

аспирант кафедры экономики промышленности и организации предприятия, Федеральное государственное бюджетное образовательное учреждение Национальный исследовательский университет «Московский энергетический институт», г. Москва, Российская Федерация, e-mail: vadikhide@yandex.ru

Аннотация. Представлены основные свойства безопасности криптографических протоколов с помощью аппарата логики линейного времени LTL, а также приведен пример автомата Бюхи, построенного на основе формулы LTL .

Ключевые слова: математическое моделирование, криптографические протоколы, защита информации, линейная темпоральная логика, формализация свойств безопасности.

FORMALIZATION OF THE SECURITY PROPERTIES OF CRYPTOGRAPHIC PROTOCOLS USING LINEAR TEMPORAL LOGIC

Vadym Borkhalenko

Postgraduate of Economics, Federal State Budget Educational Institution National Research University “Moscow Power Engineering Institute”, Moscow, Russia, e-mail: vadikhide@yandex.ru

Abstract. Conducted theoretical research of applying instruments of linear temporal logic to formalize the security properties of cryptographic protocols and developing the Buchi automata based on LTL-formulas.

Keywords: mathematical modelling, cryptographic protocols, information security, linear temporal logic, security properties formalization.

Введение. Довольно часто при разработке программного обеспечения (ПО) недостаточно времени уделяется математическому моделированию и анализу будущих программных систем. Чаще всего эта ситуация обуславливается нехваткой человеческих ресурсов и давлением по сро-

кам. Кажущаяся на первый взгляд экономия ресурсов и сокращение сроков на практике обычно приводят к противоположным результатам, особенно в случае разработки действительно сложных систем, использующих криптографические протоколы, которые представляют собой относительно сложный набор правил взаимодействия общающихся между собой сторон (процессов). Фактически в отсутствие моделей реальная производительность и адекватность программной системы могут быть проверены экспериментально, в ходе, например, опытной эксплуатации или тестирования.

Однако, у тестирования имеется ряд недостатков [1]:

- Тестирование очень трудоемкий процесс;
- Тестирование выполняется на поздних этапах разработки системы, когда модули системы уже реализованы, поэтому исправление найденных ошибок очень дорого, при исправлении вносятся часто другие ошибки;
- Все реакции системы при выполнении тестов должны быть заранее зафиксированы;
- Тестированием можно проверить лишь очень немногие траектории вычисления системы (а их обычно бесконечное множество);
- Тестирование не может гарантировать правильность работы системы: никаким количеством тестов нельзя установить правильность работы системы в любых условиях;
- Тестированием невозможно выявить редко проявляющиеся ошибки, особенно ошибки в параллельных системах.

Поэтому для более подробного анализа поведения системы необходимо прибегнуть к процессу верификации модели системы. Стандарт ИСО 9000:2000 определяет процесс верификации следующим образом:

Верификация – подтверждение на основе представления объективных свидетельств того, что установленные требования были выполнены. В отличие от тестирования, верификация позволяет проверить выполнение требований спецификации на всех возможных траекториях функционирования системы. Верификация проводится на самых ранних этапах проектирования по заранее разработанной формальной модели, что существенно экономит время и затраты на исправление ошибок. Сама же процедура проверки выполнения формальных требований на модели системы является процессом верификации.

Формальная верификация – это приемы и методы формального доказательства (или опровержения) того, что модель программной системы удовлетворяет заданной формальной спецификации [1].

Одним из методов верификации программных систем является model checking. Метод model checking основан на построении формальной модели системы (в нашем случае – криптографического протокола) с примене-

нием темпоральных логик того или иного вида. Их применение позволяет строить высказывания относительно развития процесса взаимодействия сторон в протоколе и выполнения или невыполнения определенных свойств как в течение всего сеанса, так и после его определенных этапов. Идея метода model checking заключается в построении синхронной композиции [1,3] автомата Бюхи, описывающего модель системы и автомата Бюхи, описывающего отрицание LTL-формулы, выражающей проверяемое свойство системы. Если язык, принимаемый композицией автоматов пуст, то проверяемое свойство выполняется на всех вычислениях системы.

Цель работы. Целью данного исследования является формализация некоторых из основных свойств безопасности (спецификаций) криптографических протоколов, фигурирующих в документах IETF, с помощью языка линейной темпоральной логики LTL, а также представление формул LTL в виде контрольных автоматов Бюхи.

Материал и результаты исследования. Темпоральные логики – это логики, в которых истинностное значение логических формул зависит от момента времени, в котором вычисляются значения этих формул [1].

Основная идея темпоральной логики состоит в том, чтобы фиксировать только относительный порядок событий – фактически, текущее будущее и прошлое. Современная темпоральная логика линейного времени (LTL) была разработана израильским ученым Артуром Прайом для спецификации свойств параллельных технических систем. В LTL темпоральными операторами расширена простейшая логика высказываний, формулы которой строятся из конечного числа атомарных предикатов и булевых операций.

Атомарный предикат – это утверждение, которое может принимать истинное или ложное значение и от структуры которого мы абстрагируемся. Формулы LTL принимают истинное или ложное значение на различных вычислениях реагирующей системы, т.е. на бесконечном числе последовательностей состояний, которые система проходит во времени. Поведение реагирующей системы – это все ее возможные вычисления.

В каждом состоянии может быть задан набор истинных в этом состоянии атомарных предикатов – тех базовых свойств, интересующих нас при анализе системы. Формула LTL выполнима, если она истинна на всех возможных вычислениях системы.

Формула линейной темпоральной логики это :

- Атомарные предикаты p,q;
- Логические операторы $\vee, \neg$ из которых можно выразить $\wedge, \Rightarrow$;
- Темпоральные операторы X, U из которых можно выразить F, G.

Прошлое в логике LTL не рассматривается.

Темпоральные операторы логики LTL:

- Xp – p истинно на следующем шаге;
- pUq – p истинно до тех пор, пока не будет истинно q;
- Fp – хоть раз в будущем (будущее = текущий момент и все моменты после него) будет истинно p;
- Gp - всегда истинно p.

Автомат Бюхи – это пятерка (S,Σ,S_0,δ,F)[1], где:

- Σ – конечное множество символов (алфавит);
- S – множество состояний;
- S_0 – множество начальных состояний $S_0 \in S$ (содержащее только один элемент для детерминированного случая);
- $\delta : S \times \Sigma \to 2^S$ – отношение переходов (для детерминированного автомата $\delta : S \times \Sigma \to S$);
- $F \subseteq S$ – множество допускающих (финальных состояний).

Автомат Бюхи можно считать недетерминированным конечным автоматом, получающим на вход слова бесконечной длины (ω-слова). В отличие от конечного автомата, условие допустимости входной цепочки автомата Бюхи изменено так, чтобы можно было определить некоторые бесконечные цепочки (ω-слова) как допустимые. Любое множество ω-слов над алфавитом Σ называется ω-языком над Σ.

Любое ω-слово w над алфавитом Σ можно считать функцией $w : N \to \Sigma$ (где N- натуральные числа), понимая под w(i) i-ю букву слова w. Аналогично: если $\pi : N \to S$ – бесконечная последовательность состояний автомата A, то π(i) – i-е состояние в цепочке π. Обозначим inf(π) множество элементов из S, которые встречаются в цепочке π бесконечно много раз.

Ω – слово w над алфавитом Σ допускается автоматом Бюхи $A=(S,\Sigma,S_0,\delta,F)$, если существует такое ω-слово над множеством состояний S(бесконечная цепочка состояний автомата A), что:

- $\pi(0) \in S_0$ – цепочка состояний начинается одним из начальных состояний;
- $\forall i \geq 0 : \pi(i+1) \in \delta(\pi(i), w(i))$ – переход в очередное состояние вызван очередной буквой ω-слова w;
- $\inf(\pi) \cap F \neq \varnothing$ – среди бесконечно повторяющихся состояний цепочки π существует хотя бы одно допускающее.

Таким образом, автомат Бюхи распознает (допускает) ω-слово, если при чтении этого бесконечного слова автомат бесконечно много раз проходит хотя бы одно допускающее состояние. Множество ω-слов, допускаемых автоматом Бюхи A, называется ω-языком A, допускаемым A. ω-язык $L \subseteq \Sigma^{\omega}$ называется Бюхи-допускаемым, если существует автомат Бюхи, допускающий L.

Формальное представление свойств безопасности криптографических протоколов. Обычно свойства протоколов, характеризующие их стойкость к различным атакам, формулируют как цели или требования к протоколам. Наиболее полное и совершенное толкование этих целей дается в документах международной организации Internet Engineering Task Force (ITEF). Рассмотрим и формализуем некоторые из них [2]:

Свойство защиты от повтора – это гарантирование одним из участников того, что аутентифицированное сообщение не является старым:

$$G((chan!mes\_Y \Rightarrow FG\neg(chan!mesY) \wedge F(chan!mes\_Y)) \quad (1)$$

Где chan!mes_Y – атомарный предикат, описывающий событие отправки сообщения mes_Y в канал связи chann.

(Если сообщение Y послано в канал, то оно никогда в будущем не пошлет такое же сообщение).

Конечно, в данном случае рациональнее было бы использовать совокупность операторов XG вместо FG. Однако темпоральный оператор X не может быть использован при построении LTL формул для верификации параллельных систем, поскольку понятия «следующего состояния» в системе параллельных процессов нет: в общем случае каждый из параллельных процессов может выполнить шаг.

Свойство аутентификации субъекта – это проверка с подтверждением подлинности одной из сторон наличия или полномочий идентичности второй стороны, участвующей в выполнении протокола, а также того, что она действительно принимает участие в выполнении текущего сеанса протокола. Таким образом, обычно аутентификация субъекта подразумевает, что некоторые данные могут быть безошибочно возвращены некоторому субъекту, что предполагает аутентификацию источника:

$$G(chan!A\_response\_B \Rightarrow A\_commit\_Na') \quad (2)$$

А всегда отправит подтверждение в канал, если А подтвердит подлинность своего принятого нонса.

Свойство подтверждения ключа – один из участников получает подтверждение того, что второй участник действительно обладает секретным ключом.

$$G(((chan!\{K1\}_K \wedge B\_Knows\_K) \Rightarrow F(chan?\{B\_response\}_{K1} \wedge B\_Knows\_K1)) \wedge F(chan!\{K1\}_K \wedge B\_Knows\_K) \quad (3)$$

Свойство конфиденциальности состоит в том, что специфический набор данных (обычно посылаемый или получаемый как часть «защищенного» сообщения, а также сформированный на данных, полученных в результате обмена) не станет доступным или раскрытым для неавторизированных субъектов или процессов, а останется неизвестным противнику.

Выразим данное свойство с помощью классического свойства безопасности системы:

$$\neg F(A\_Knows\_K_{AB} \wedge B\_Knows\_K_{AB} \wedge I\_Knows\_K_{AB}) \quad (4),$$

где I – нарушитель.

Приведем пример контрольного автомата Бюхи, построенного на основе спецификации (2).

Построим отрицание свойства (2):

$$\begin{aligned} &\neg G(chan!A\_response\_B \Rightarrow A\_commit\_Na') = \\ &= F(chan!A\_response\_B \wedge \neg A\_commit\_Na') \end{aligned} \quad (5)$$

На рисунке 1 приведен автомат, построенный по заданному отрицанию [1, 4]:

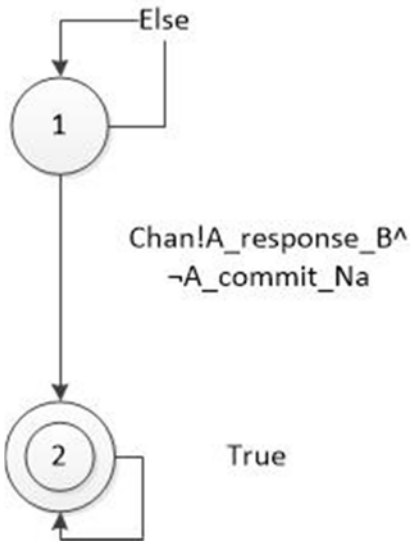

Рисунок 1 – Автомат, описывающий формулу $\neg G(chan!A\_response\_B \Rightarrow A\_commit\_Na')$

Данный автомат можно использовать, как контрольный для автомата, описывающего модель системы (в нашем случае криптографического протокола). Если язык, принимаемый синхронной композицией автомата си-

стемы и контрольного автомата, будет пуст, то проверяемое свойство выполняется на всех вычислениях системы.

Вывод. В результате работы были формализованы с помощью языка логики LTL следующие свойства безопасности: защита от повтора, аутентификация субъекта, подтверждение ключа и конфиденциальность. Также на основе одной из полученных формул был построен автомат Бюхи, который является контрольным для анализируемой системы. Дальнейшей задачей автора является формализация свойств протоколов электронной коммерции, разработка модели нарушителя и верификация одного из протоколов электронной торговли с применением модели нарушителя.

ЛИТЕРАТУРА

1. Карпов Ю.Г. Model Checking. Верификация параллельных и распределенных систем. – БХВ-Петербург, 2009.

2. Черемушкин А.В. Криптографические протоколы: основные свойства и уязвимости. – М.: Изд-во Academia, 2009.

3. Соколов А. П. Системы программирования. – М.: Изд-во Финансы и Статистика, 2004.

4. Krawczyk U., Sapiecha P. Effective reduction of cryptographic protocols specification of model-checking with Spin / Krawczyk U., Sapiecha P. // Annales UMCS Informatica. – Lublin, 2011. – № 3.– С. 27 – 40.

УДК 044.421: 624.074.435

АЛГОРИТМ ВИЗНАЧЕННЯ НДС ТОНКОЇ КОНІЧНОЇ ОБОЛОНКИ ПРИ ОДНОБІЧНІЙ В'ЯЗІ МІЖ НЕЮ ТА ПРУЖНОЮ ОСНОВОЮ

О.В. Запорожець[1], С.М. Горлач[2], В.Б. Запорожець[3]

[1]кандидат технічних наук, доцент кафедри прикладної математики, Державний вищій навчальний заклад «Придніпровська державна академія будівництва та архітектури», м. Дніпропетровськ, Україна, e-mail: Lena_ne@ukr.net

[2]кандидат технічних наук, доцент кафедри основ і фундаментів, Державний вищій навчальний заклад «Придніпровська державна академія будівництва та архітектури», м. Дніпропетровськ, Україна, e-mail: serg.gorlach@ukr.net

[3]кандидат технічних наук, доцент кафедри будівельної механіки та опору матеріалів, Державний вищій навчальний заклад «Придніпровська державна академія будівництва та архітектури», м. Дніпропетровськ, Україна

Анотація. Наведено алгоритм визначення напружено-деформованого стану тонкої конічної оболонки, яка розташована на пружній основі. Між конічною оболонкою та основою може бути як двобічна, так і однобічна в'язі . Ця задача розв'язується за допомогою метода скінченних елементів. У випадку однобічної в'язі переміщення конічної оболонки визначаються за допомогою метода послідовних наближень. За представле-