ФИЗИКО-МАТЕМАТИЧЕСКИЕ НАУКИ

УДК 004.414

Программное средство логической проверки корректности криптографических протоколов распределения ключей на основе BAN-логики

Н. С. Могилевская
(Донской государственный технический университет)

Рассмотрена идея анализа криптографических протоколов распределения ключей методами BAN-логики; приведён пример анализа протокола Керберос; построено программное средство, автоматизирующее анализ протоколов распределения ключей; исследована корректность его работы.
Ключевые слова: *формальный анализ протоколов, криптографические протоколы, протоколы распределения ключей, средства автоматизированного анализа протоколов, протокол Керберос.*

Введение. Криптографические протоколы играют значимую роль в защите информации. В разное время было создано большое количество криптографических протоколов различного назначения. Фактически протокол представляет собой список сообщений, которыми должны обменяться участники протокола. Несмотря на кажущуюся простоту, проблема оценки уровня безопасности протоколов является очень сложной. Так многие протоколы, считавшиеся надёжными долгое время, спустя десятки лет демонстрировали свою уязвимость в результате атак [1, 2]. Если определять криптографический протокол более строго, то это — распределённый алгоритм последовательности шагов, точно специфицирующих действия, которые требуются от участников для решения некоторой криптографической задачи [1, 3].

Для исследования уровня безопасности, который может обеспечить как уже существующий, так и только разрабатываемый протокол, используют специализированные формальные методы. Среди математических аппаратов, используемых для решения задачи формального анализа протокола, важную роль играют различные логики, среди которых наибольшее распространение получили логики доверия. В работе [4] была предложена BAN-логика, которая стала основой для разработки ряда других логик. Данная логика является первой попыткой построения формального языка для описания исходных предположений, правил вывода и конечных целей анализа безопасности. BAN-логика, как и её расширения, позволяют проводить анализ протокола вручную, однако это достаточно трудоёмко и чревато ошибками. Целесообразным представляется использовать автоматизированные системы анализа, но в настоящее время подобных программ в свободном доступе нет. Программные средства (ПС), реализующие анализ протоколов методами BAN-логики, могут быть полезны и для тестирования работоспособности и безопасности криптографических протоколов обмена ключами, и для учебных целей, так как изучение любых логик доверия базируется на владении навыками анализа методами BAN-логики.

Постановка задачи. Создать ПС BanAnalyzer, автоматизирующее процесс анализа криптографических протоколов методами BAN-логики. Входные и выходные данные программного средст-

ва должны быть максимально приближены по своему формату к идеализированным протоколам, предлагаемым создателями BAN-логики в оригинальной работе [4].

Для выполнения задачи кратко опишем идеи, лежащие в основе BAN-логики, и сконструируем алгоритмы работы ПС и опишем идею его создания. Затем рассмотрим пример анализа протокола с использованием ПС BanAnalyzer и исследуем корректность результатов его работы.

Основные положения BAN-логики. Данная логика используется для анализа протоколов распределения ключей [4—6]. Основная идея BAN-логики состоит в отслеживании восприятия сторонами поступающей информации, а именно: какие данные они принимают на веру, какие данные им точно известны и какие могут быть выведены логическим путём из достоверных для них фактов. Так, для каждого шага протокола методами BAN-логики формируется список утверждений о безопасности протокола, которым доверяют участники протокола.

При использовании BAN-логики не моделируются ни различие между простым просмотром сообщения и пониманием его, ни пересмотр доверий, ни знание. Все эти аспекты адресуются к неформальному отображению спецификации протокола в спецификацию BAN-логики, которое авторы [4] называют идеализацией. Таким образом, анализу протокола предшествует его идеализация, которая производится человеком самостоятельно согласно описанию протокола. Идеализированные протоколы считаются более ясными и обладающими более полной спецификацией, чем традиционные описания. В терминах BAN-логики протокол рассматривается на абстрактном уровне, следовательно, ошибки конкретной реализации, такие как тупики или неправильное использование криптосистемы, при анализе не обнаруживаются. Методами BAN-логики анализируется непосредственно криптографический протокол и его логика, а используемые в нём криптографические методы считаются стойкими.

Перечислим объекты, которые различают в BAN-логике, и укажем их обозначения. Участники протокола обычно обозначаются A, B, S; общие ключи шифрования, применяемые при симметричной криптографии, обозначаются K_{AB}, K_{AS} и K_{BS}, где индексы в обозначении указывают на участников, использующих данных ключи; открытые ключи, используемые при асимметричной криптографии, обозначаются K_A, K_S и K_B (где A, B, S — участники, которым принадлежат данные открытые ключи), связанные с ними секретные ключи обозначаются K_A^{-1}, K_S^{-1} и K_B^{-1} соответственно; N_A, N_B, N_S — специальные числовые значения (нонсы, метки времени); X, Y — общее обозначение для формул и утверждений.

Единственная используемая логическая операция в BAN-логике — конъюнкция обозначается запятой. Свойства ассоциативности и коммутативности считаются доказанными.

Укажем базовую систему обозначений, принятую в BAN-логике.

$P \mid\equiv X$ — участник протокола P верит (believes) утверждению X; далее участник P будет действовать, считая, что утверждение X верно.

$P \triangleleft X$ — участник P видит (sees) утверждение X; участник P получил от кого-то утверждение X и может его прочитать и повторить.

$P \mid\sim X$ — участник P однажды заявил (once said) утверждение X, и в тот момент P верил утверждению X, однако время этого высказывания неизвестно.

$P \mid\Rightarrow X$ — P обладает полномочиями (jurisdiction) над X; т. е. участник P является автором утверждения X и верит в него. Эта конструкция часто обозначает, что пользователь имеет права на создание ключей.

$\#(X)$ — утверждение X является свежим (fresh). Под термином «свежий» понимается, что утверждение X сгенерировано в текущем сеансе связи и не было послано до начала работы протокола.

$P \overset{K}{\leftrightarrow} Q$ — участники P и Q могут использовать общий ключ K для установления связи. Предполагается, что ключ K достаточно стоек и не может быть взломан кем-либо из посторонних, если это не предусмотрено протоколом.

$\overset{K}{\rightarrow} P$ — P имеет открытый ключ K (public key), а также согласованный с ним качественный секретный ключ K^{-1}, никому не известный, кроме P или участника, которому он доверяет.

$P \overset{X}{\Leftrightarrow} Q$ — утверждение X является секретом, известным только участникам P и Q, и они могут использовать X для доказательства своей аутентичности один другому.

$\{X\}_K$ — данные X зашифрованы с использованием ключа K. Шифрование считается надёжным.

$\langle X \rangle_K$ — конкатенация утверждения X и секрета Y. Секрет Y полностью идентифицирует объект, заявивший утверждение X.

При анализе протоколов аутентификации различают два времени: прошлое и настоящее. Настоящее время начинается со старта данного сеанса работы протокола. Все сообщения, посланные до этого, считаются старыми сообщениями, и в ходе работы протокола необходимо предотвращать возможность появления таких сообщений. Все веры, принятые в настоящем, неизменны на протяжении всего сеанса работы протокола, однако те веры, которые были приняты в прошлом, не обязательно должны быть переведены в настоящее. Такое простое разделение времени на прошлое и настоящее является достаточным для использования в BAN-логике.

Ниже укажем наиболее важные правила вывода BAN-логики, используемые для получения новых утверждений и доверий участников протокола. В постулатах используем запись вида

$$\frac{A, B}{C},$$

что означает, что так как утверждения A и B верны, то верно и утверждение C.

Первые три выражения задают так называемые правила значения сообщений. Основное их различие состоит в том для получения одного и того же утверждения используются различные исходные веры. Два первых правила позволяют интерпретировать шифрованные сообщения, а третье правило позволяет интерпретировать сообщения с секретами. Они все объясняют процесс получения верований о происхождении сообщений:

$$\frac{P \mid\equiv Q \overset{K}{\leftrightarrow} P, P \lhd \{X\}_K}{P \mid\equiv Q \mid\sim X}, \qquad (1)$$

$$\frac{P \mid\equiv \overset{K}{\rightarrow} Q, P \lhd \{X\}_{K^{-1}}}{P \mid\equiv Q \mid\sim X},$$

$$\frac{P \mid\equiv Q \overset{Y}{\Leftrightarrow} P, P \lhd \{X\}_{K^{-1}}}{P \mid\equiv Q \mid\sim X}.$$

Дадим эквивалентную словесную формулировку первому из этих выражений: из предположений о том, что P верит в совместное с Q использование ключа K, и P видит сообщение X, зашифро-

ванное ключом K, делаем вывод: P верит, что Q в какой-то момент высказал X. Заметим, что здесь неявно предполагается, что сам P никогда не высказывал X.

Правило проверки нонсов:

$$\frac{P \mid\equiv \#(X), P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}, \tag{2}$$

т. е. если P верит, в свежесть сообщения X и верит, что Q когда-то высказал X, то он верит в то, что Q по-прежнему доверяет X.

Правило полномочий:

$$\frac{P \mid\equiv Q \mid\Rightarrow X, P \mid\equiv Q \mid\equiv X}{P \mid\equiv X} \tag{3}$$

говорит, что вера P в полномочия Q относительно X и вера Q в X, влекут за собой веру P в X.

Как уже было сказано, перед непосредственным анализом протокола его необходимо представить в идеализированной форме. Для этого нужно записать шаги протокола в терминах BAN-логики. Обычно в литературе каждый шаг протокола записывается в виде символьной строки:

$$P \rightarrow Q : \textit{сообщение}.$$

Такая запись означает, что участник P посылает сообщение участнику Q, а Q получает это сообщение. Сообщение представляет собой строчку, содержащую различные данные. Сообщение в идеализированном протоколе — это формула. Например, в описании протокола может быть такая символьная запись:

$$A \rightarrow B : \{A, K_{AB}\}_{K_{BS}},$$

которая означает, что B получил сообщение от A. Сообщение зашифровано ключом K_{BS} связи участника B и доверенного сервера S и содержит имя участника протокола A, а также ключ K_{AB} для связи участников A и B. Этот шаг может быть идеализирован как:

$$B \lhd \left\{ A \overset{K_{AB}}{\leftrightarrow} B \right\}_{K_{BS}},$$

что означает, что участник B принял сообщение и может действовать дальше на основе полученных данных. В идеализированной форме опускаются части сообщений, которые не способствуют получению новых формул. Например, можно опустить сообщения, используемые как рекомендации о необходимости инициализации связи, то есть, как будто участники действуют спонтанно. Идеализированные протоколы не включают открытый текст как часть сообщения, так как эти части могут быть подделаны.

Идеализированные протоколы считаются более ясными и более законченными спецификациями, чем традиционные описания, используемые в литературе, поэтому авторы BAN-логики рекомендуют использовать идеализированные формы при изобретении и описании протоколов. Получение практического вида протокола из идеализированной формы, хотя и не совсем тривиально, но менее трудоёмко и менее подвержено ошибкам, чем однозначное понимание специфических неформальных записей протоколов. К сожалению, идеализация протокола производится человеком самостоятельно, этот этап нельзя автоматизировать, и, следовательно, при его выполнении возможны ошибки. К тому же не существует строгого алгоритма записи протокола в идеализированной форме. Более того, в ряде случаев идеализация одного и того же шага протокола может быть выполнена различными способами.

Укажем в общем виде утверждения, достижение которых обычно и является целью анализа протокола с использованием симметричной криптографии.

Для протоколов передачи ключей минимальными целями являются:

$$A \mid\equiv A \overset{K}{\leftrightarrow} B,\ B \mid\equiv A \overset{K}{\leftrightarrow} B$$

т. е. оба участника верят в то, что у них есть общий ключ K для связи. Однако можно потребовать от протокола большего, например уверенности участников в свежести ключа:

$$A \mid\equiv \#\left(A \overset{K}{\leftrightarrow} B\right), B \mid\equiv \#\left(A \overset{K}{\leftrightarrow} B\right),$$

а также уверенности каждого из них в том, что другой участник также верит в этот ключ:

$$A \mid\equiv B \mid\equiv A \overset{K}{\leftrightarrow} B,\ B \mid\equiv A \mid\equiv A \overset{K}{\leftrightarrow} B.$$

Такие утверждения называют подтверждением приёма ключа. Т. е. в результате работы протокола A будет уверен в знании B о том, что он разделяет секретный ключ с A, а B будет верить в то, что A знает об их общем ключе. В случае использования асимметричной криптографии цели анализа протокола формируются аналогичными утверждениями.

Пример анализа протокола методами BAN-логики. Рассмотрим хорошо известный протокол Керберос [2, 3], разработанный как часть проекта Project Athena корпорацией MIT. Протокол позволяет двум участникам, используя доверенный сервер аутентификации, получить общий ключ. Напомним, что протокол состоит из 4 шагов. На первом участник A посылает серверу S сообщение, указывая участника B, с которым хочет установить общий ключ; на втором шаге сервер S отправляет A новый ключ K_{AB}, зашифрованный ранее установленным между A и S общим ключом K_{AS}, а также K_{AB}, зашифрованный общим между B и S ключом K_{BS}. В своё сообщение S добавляет временные метки для подтверждения свежести ключа; на следующем шаге участник A пересылает участнику B сообщение из двух частей: ключ K_{AB}, зашифрованный K_{BS}, а так же ключ K_{AB}, зашифрованный этим же ключом. В своё сообщение A добавляет временные метки; на последнем шаге участник B пересылает A сообщение со своей временной меткой, закрытое новым общим ключом K_{AB}. В традиционной символьной записи шаги этого протокола выглядят следующим образом:

$$\textit{Шаг}\ 1: A \to S: \{A, B\}$$

$$\textit{Шаг}\ 2: S \to A: \left\{T_s, K_{AB}, \{T_s, K_{AB}, A\}_{K_{BS}}\right\}_{K_{AB}}$$

$$\textit{Шаг}\ 3: A \to B: \{T_s, K_{AB}, A\}_{K_{BS}}, \{T_A, K_{AB}\}_{K_{AB}}$$

$$\textit{Шаг}\ 4: B \to A: \{T_A - 1, K_{AB}\}_{K_{AB}}$$

Построим идеализацию протокола. Отметим, что первый шаг отбрасывается, так как не имеет ценности для дальнейшего анализа, а служит только для инициализации протокола:

$$\textit{Шаг}\ 2: A \lhd \left\{T_s, \left(A \overset{K_{AB}}{\leftrightarrow} B\right), \left\{T_s, \left(A \overset{K_{AB}}{\leftrightarrow} B\right)\right\}_{K_{BS}}\right\}_{K_{AS}}$$

$$\textit{Шаг}\ 3: B \lhd \left\{T_S, \left(A \overset{K_{AB}}{\leftrightarrow} B\right)_{K_{BS}}, \left(T_a, A \overset{K_{AB}}{\leftrightarrow} B\right)_{K_{AB}}\right\}$$

$$\textit{Шаг } 4: \left\{ T_A, \left(A \overset{K_{AB}}{\leftrightarrow} B \right) \right\}_{K_{AB}}$$

В табл. указаны первоначальные доверия участников, необходимые для начала работы протокола.

Доверия участников протокола Керберос (в терминах BAN-логики)

Доверия	Значение
$A \mid\equiv A \overset{K_{AS}}{\leftrightarrow} S$	*A* верит, что между *A* и *S* установлен общий ключ для симметричного шифрования K_{AS}
$B \mid\equiv B \overset{K_{BS}}{\leftrightarrow} S$	*B* верит, что между *B* и *S* установлен общий ключ для симметричного шифрования K_{BS}
$A \mid\equiv S \Rightarrow K_{AB}$	*A* доверяет серверу аутентификации *S* генерацию симметричного ключа K_{AB}
$B \mid\equiv S \Rightarrow K_{AB}$	*B* доверяет серверу аутентификации генерацию симметричного ключа K_{AB}
$A \mid\equiv \#T_S$	*A* верит в то, что временная вставка T_S, созданная *S*, актуальна (доказывает свежесть сообщения от *S*)
$B \mid\equiv \#T_S$	*B* верит в то, что временная вставка T_S, созданная *S*, актуальна (доказывает свежесть сообщения от *S*)
$A \mid\equiv \#T_B$	*A* верит в то, что временная вставка *B* актуальна (доказывает свежесть сообщения от *B*)
$B \mid\equiv \#T_A$	*B* верит в то, что временная вставка *A* актуальна (доказывает свежесть сообщения от *A*)

Проанализируем шаги протокола последовательно один за другим, по возможности применяя к ним все основные правила BAN-логики. Запишем основные значимые этапы анализа. На втором шаге, применяя правило значения сообщений (1) получаем:

$$\frac{A \mid\equiv A \overset{K_{AS}}{\leftrightarrow} S, A \triangleleft \left\{ T_S, \left(A \overset{K_{AB}}{\leftrightarrow} B \right), \left\{ T_S, \left(A \overset{K_{AB}}{\leftrightarrow} B \right) \right\}_{K_{BS}} \right\}_{K_{AS}}}{A \mid\equiv S \mid\sim T_S, \left(A \overset{K_{AB}}{\leftrightarrow} B \right), \left\{ T_S, \left(A \overset{K_{AB}}{\leftrightarrow} B \right) \right\}_{K_{BS}}}. \quad (4)$$

Из правила проверки нонсов (2) получаем

$$\frac{A \mid\equiv S \mid\sim T_s, \left(A \overset{K_{AB}}{\leftrightarrow} B \right), \left\{ T_s, \left(A \overset{K_{AB}}{\leftrightarrow} B \right) \right\}_{K_{BS}}, A \mid\equiv \#T_s}{A \mid\equiv S \mid\equiv \left(A \overset{K_{AB}}{\leftrightarrow} B \right)}. \quad (5)$$

Применение правила проверки полномочий (3) даёт нам следующий результат:

$$\frac{A \mid\equiv S \mid\equiv \left(A \overset{K_{AB}}{\leftrightarrow} B \right), A \mid\equiv S \mid\Rightarrow \left(A \overset{K_{AB}}{\leftrightarrow} B \right)}{A \mid\equiv \left(A \overset{K_{AB}}{\leftrightarrow} B \right)}. \quad (6)$$

Таким образом, в результате шага 2 протокола участник *A* доверяет полученному от *S* ключу для связи *A* и *B*.

Последовательно применяя (1), (2) и (3) к шагу 3 протокола получаем

$$B \mid\equiv A \overset{K_{AB}}{\leftrightarrow} B, \; B \mid\equiv A \mid\equiv A \overset{K_{AB}}{\leftrightarrow} B.$$

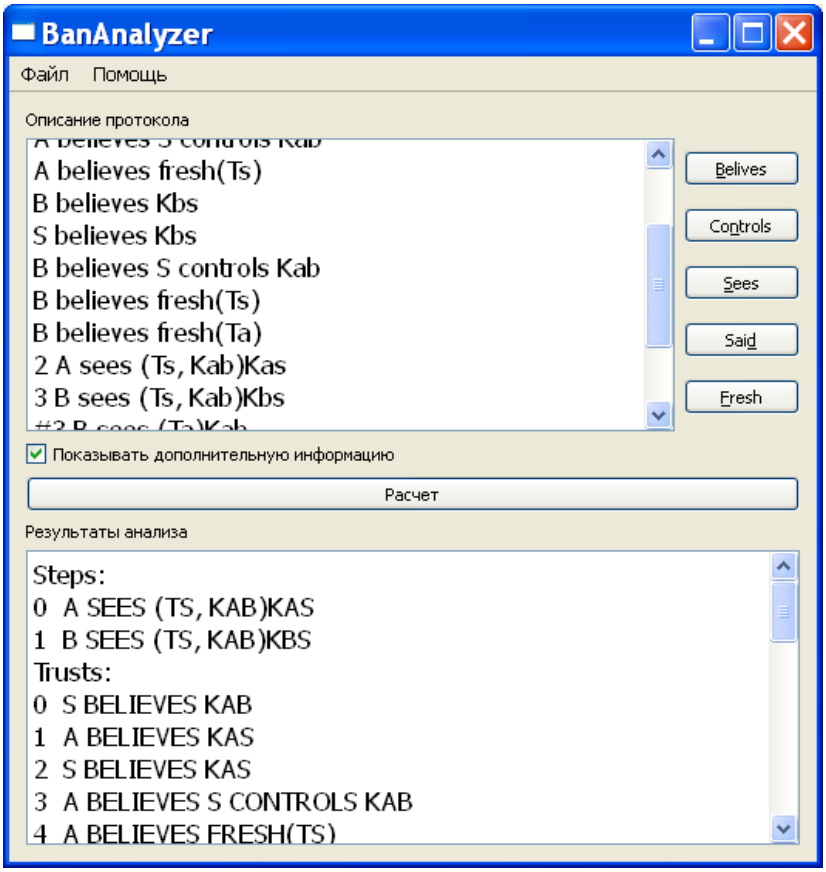
Результат анализа шага 4 протокола даёт формулу:

$$A \mid\equiv B \mid\equiv A \overset{K_{AB}}{\leftrightarrow} B.$$

Таким образом, анализ BAN-логикой показал, что выполнение протокола Керберос обеспечивает достижение всех целей симметричного протокола обмена ключами без использования дополнительных доверий. Ещё раз подчеркнём, что использованная логика не показывает уязвимостей протокола, связанных со слабостью процедур шифрования и ошибками реализации. По результатам анализа мы получили знание о логической корректности протокола.

Алгоритмическое конструирование ПС. Алгоритмы работы программы условно можно разделить на четыре части: алгоритмы, реализующие взаимодействие с пользователем; алгоритм разбора введённого пользователем текста описания протокола на утверждения BAN-логики; алгоритм анализа протокола и алгоритм формирования выходных данных.

Алгоритм разбора пользовательского ввода на утверждения BAN-логики получает на вход текст описания протокола. Текст может содержать шаги протокола, принятые доверия и комментарии к описанию. Комментарии из дальнейшего анализа исключаются как незначимые для анализа протокола. Выделить их в описании достаточно просто по служебному символу «#», предваряющему текст комментария. Шаги протокола отличаются от доверий тем, что их запись начинается с номера. Затем в строках, содержащих шаги и доверия, выделяем имена участников, ключевые слова (believes, sees, said, controls) и т. д. Алгоритмом по мере работы заполняются списки шагов, доверий и сообщений об ошибках — эти три списка считаются выходными данными этого алгоритма.



S believes Kab
A believes Kas
S believes Kas
A believes S controls Kab
A believes fresh(Ts)
B believes Kbs
S believes Kbs
B believes S controls Kab
B believes fresh(Ts)
B believes fresh(Ta)
2 A sees (Ts, Kab)Kas
3 B sees (Ts, Kab)Kbs
#3 B sees (Ta)Kab
#4 A sees (Ta+1)Kab

Рис. 1. Основное окно ПС BanAnalyzer и полный текст описания исследуемого протокола Керберос

Наиболее трудоёмким является алгоритм проведения анализа. Он различается для симметричных и асимметричных криптографических протоколов. Оба варианта алгоритма включа-

ют в себя как составную часть алгоритм поиска по довериям и результатам. Результат работы алгоритма поиска указывает, известно ли уже данное утверждение и чем оно является — первоначальным доверием или результатом анализа. Анализ каждого из шагов протокола проходит через три стадии, на каждой из которых последовательно применяются основные правила BAN-логики: проверка значений сообщений, проверка нонсов и проверка полномочий, при необходимости к основным правилам добавляется ещё ряд стандартных правил BAN-логики, не описанных в данной работе.

Выходные данные представляются в виде текстового файла, который формируется в следующем порядке: сначала к выходному тексту добавляем все распознанные шаги протокола в естественном порядке, затем добавляем все распознанные доверия протокола, после этого дописываем все полученные результаты анализа протокола в порядке их получения, в заключение вносим в результирующий файл информацию о всех неиспользованных довериях протокола, записи о правилах BAN-логики, применённых к каждому из шагов протокола, записи об ошибках во время распознавания шагов и доверий протокола.

Программное конструирование анализатора BAN-логики BanAnalyzer. Для реализации программного средства был использован высокоуровневый язык программирования C++ в совокупности с бесплатным фреймворком Qt4. Архитектурный каркас данного ПС реализует шаблон проектирования MVC («модель — представление — контроллер»). Это подразумевает, что модели данных приложения, пользовательский интерфейс и основная логика работы приложения разделены на три слабосвязных компонента так, что модификация одного из компонентов оказывает минимальное воздействие на остальные.

Роль обязательной для C++ функции main() в разработанном ПС заключается в создании главного окна приложения и запуска цикла обработки событий. Модель архитектуры MVC реализована в модуле models, представленном файлами models.h и models.cpp. Основой программы является класс Expression, представляющий в программе выражения BAN-логики. Класс Exression имеет три поля: строковое поле who; поле type, принимающее одно из четырёх возможных значений: BELIEVES, CONTROLS, SAID, SEES, и поле what, являющееся объектом класса Subject, определённого в этом же модуле. Класс Expression имеет два конструктора; метод toQString, возвращающий строковое представление выражения, и статический метод stringIsType(QString), возвращающий значение «истина», если параметр функции является допустимым значением для поля type, и «ложь» в противном случае. Класс BanAnalyzer, унаследованный от QMainWindow, — основная составная часть представлений модели MVC. Важнейшими методами класса являются следующие. Input_parsing() реализует разбор пользовательского ввода на выражения BAN-логики, создание объектов класса Expression и добавление их к собственному объекту класса BanControl. On_pushButton_clicked() — обработчик события нажатия на кнопку «Расчёт», запускает метод input_parsing(), затем метод контроллера для процедуры анализа, после чего формирует окончательный вывод результата. Обработчики нажатия на кнопки конструктора описаний, добавляют соответствующие им слова в поле для ввода в текущую позицию курсора. Обработчики вызовов меню. Класс HelpDialog, унаследованный от QDialog, реализует диалоговое окно помощи. Основными функциональными компонентами являются следующие. Метод showHelp(QString) позволяет отображать текстовое содержание файла с именем, указанным в параметре на поле вывода диалогового окна. Обработчики нажатий на кнопки диалога вызывают метод showHelp, передавая ему различные имена файлов, в которых хранятся соответствующие справочные тексты. Класс BanControl — контроллер программы по архитек-

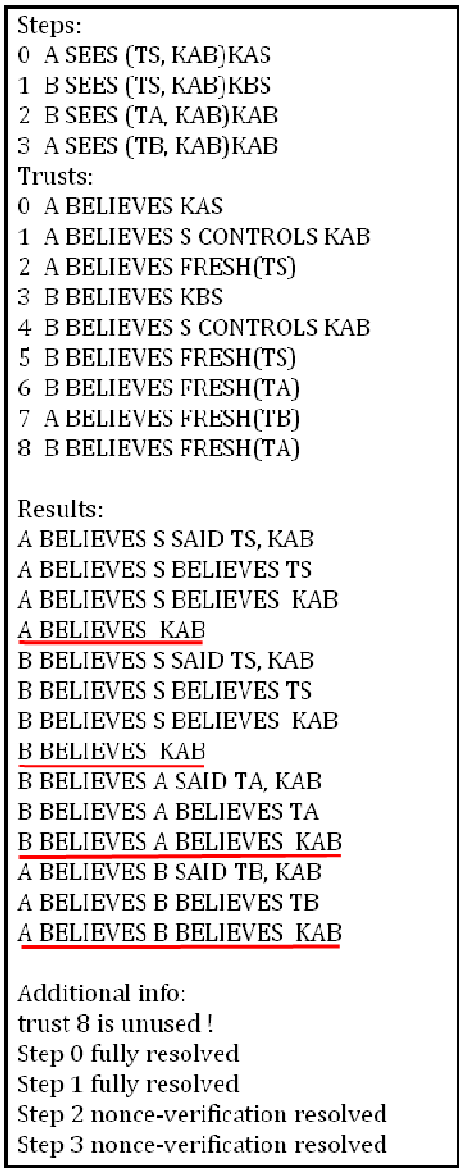


Рис. 2. Результат автоматизированного анализа протокола Керберос

type MVC, инкапсулирует в себе динамические списки шагов (steps), доверий (trusts) и результатов (results) протокола, список строк со сведениями об ошибках (messages) и некоторую служебную информацию, в том числе сведения об использовании шагов (массив steps_analisys) и доверий (массив trust_using). Алгоритм процедуры анализа протокола BAN-логикой выполняется в теле метода calculating().

В левой части рис. 1 приведён скриншот основного окна ПС BanAnalyzer, содержащий описание протокола Керберос и результаты его анализа. В программе для записи шагов протоколов и доверий участников использованы обозначения BAN-логики, предложенные разработчиками BAN-логики. Кнопки Believes, Controls, Said, Sees, Fresh используются для вставки соответствующих кванторов при формировании описания протокола.

Пример проведения анализа протокола с использованием ПС BanAnalyzer. На правой части рис. 1 представлено полное описание протокола Керберос, введённое в программу для анализа. Результат анализа протокола Керберос с помощью построенного ПС представлен на рис. 2. Напомним, что по результатам ручного анализа протокол Керберос обеспечил достижение четырёх целей симметричного протокола обмена ключами, в результате автоматизированного анализа достигнуты те же цели. Итоговые результаты на рис. 2 подчёркнуты.

Исследование работоспособности ПС BanAnalyzer. Для испытания качества разработанного ПС были подготовлены идеализированные описания ряда из четырнадцати хорошо известных протоколов (Керберос, асимметричного и симметричного протоколов Нидхема — Шрёдера, Ньюмана — Стабблайна, Ву — Лама, Деннинга — Сакко, Отвея — Рииса, Andrew RPC Handshake, DASS, BAN-Yahalom, Station-to-station, EKE, SPX, «Широкоротая лягушка» [2, 6, 7]). Для каждого из этих протоколов был проведён анализ «вручную» и подготовлен список известных из литературы возможных атак. В ходе испытаний проводился автоматизированный анализ с помощью разработанного ПС и затем сравнивались результаты автоматизированного, ручного анализа и известных уязвимостей. Проведённые испытания показали, что реализованное ПС BanAnalyzer корректно выполняет формальный анализ криптографических протоколов распределения ключей методами BAN-логики и может быть использовано как для изучения уже существующих протоколов, так и в процессе разработки новых протоколов для предотвращения их возможных уязвимостей.

Заключение. В работе построено автоматизированное ПС, позволяющее проводить формальную проверку логической корректности криптографических протоколов распределения ключей на основе использования BAN-логики. Проведённые исследования показали корректность его

работы. В настоящее время построенное ПС доступно во внутренней сети ФГБОУ ВПО «ДГТУ» для использования студентами специальности 090103 «Компьютерная безопасность» в рамках изучения дисциплины «Криптографические протоколы» [5].

В качестве дальнейшего направления работы представляется интересным построить программные реализации для других логик доверия, например, AUTLOG (V. Kessler, G. Wedel), логики объяснений (R. Kaylar), RV-логики (D. Kindred), GNY (L. Gong, R. Needham, R. Yahalom), BGNY/HOL, SvO (P. Syverson), что позволит легко анализировать протоколы этими методами, а также проводить сравнительный анализ как различных протоколов, так и результатов анализа одного и того же протокола различными логиками. Считаем важной задачей организацию свободного доступа к разработанной программе и её исходным кодам через Интернет для всех заинтересованных в использовании, изучении, а также дальнейшей разработке.

Библиографический список

1. Могилевская, Н. С. Верификация криптографических протоколов распределения ключей с использованием раскрашенных сетей Петри / Н. С. Могилевская, С. С. Колчанов // Вестник Донского гос. техн. ун-та. — 2011. — Т. 11. — № 9. — С. 1535—1543.

2. Черёмушкин, А. В. Криптографические протоколы: основные свойства и уязвимости / А. В. Черёмушкин. — Москва: Ин-т криптографии, 2009. — 272 с.

3. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. — Москва: Триумф, 2002. — 816 с.

4. Burrows, M. A logic of authentication / M. Burrows, M. Abadi, R. Needham // ACM Transactions on Computer System. — V. 8. — № 1. — Feb. 1990. — P. 18—36.

5. Могилевская, Н. С. Основы BAN-логики: метод. указания к практическим занятиям по курсу «Криптографические протоколы» [Электрон. ресурс] / Н. С. Могилевская. — Режим доступа: http://de.dstu.edu.ru/CDOCourses/3/3/20125c3d5375-aa2f-41fe-ac35-b71dc060ae20/1001/method/index.html (дата обращения: 15.10.2011).

6. Могилевская, Н. С. Сравнение возможностей сетей Петри и BAN-логики в анализе криптографических протоколов проверки подлинности и обмена ключами / Н. С. Могилевская, С. С. Колчанов // Системный анализ, управление и обработка информации. — Ростов-на-Дону: Изд. центр ДГТУ, 2011. — С. 98—101.

7. Aly, S. Protocol verification and analysis using colored Petri nets. Technical report / S. Aly. — Cairo: Cairo University, 2003. — 26 p.

Материал поступил в редакцию 02.12.2011.

References

1. Mogilevskaya, N. S. Verifikaciya kriptograficheskix protokolov raspredeleniya klyuchej s ispol`zovaniem raskrashenny`x setej Petri / N. S. Mogilevskaya, S. S. Kolchanov // Vestnik Donskogo gos. texn. un-ta. — 2011. — T. 11. — № 9. — S. 1535—1543. — In Russian.

2. Cheryomushkin, A. V. Kriptograficheskie protokoly`: osnovny`e svojstva i uyazvimosti / A. V. Cheryomushkin. — Moskva: In-t kriptografii, 2009. — 272 s. — In Russian.

3. Shnajer, B. Prikladnaya kriptografiya. Protokoly`, algoritmy`, isxodny`e teksty` na yazy`ke Si / B. Shnajer. — Moskva: Triumf, 2002. — 816 s. — In Russian.

4. Burrows, M. A logic of authentication / M. Burrows, M. Abadi, R. Needham // ACM Transactions on Computer System. — V. 8. — № 1. — Feb. 1990. — P. 18—36.

5. Mogilevskaya, N. S. Osnovy` BAN-logiki: metod. ukazaniya k prakticheskim zanyatiyam po kursu «Kriptograficheskie protokoly`» [E`lektron. resurs] / N. S. Mogilevskaya. — Rezhim dostupa:

http://de.dstu.edu.ru/CDOCourses/3/3/20125c3d5375-aa2f-41fe-ac35-b71dc060ae20/1001/method/index.html (data obrashheniya: 15.10.2011). — In Russian.

6. Mogilevskaya, N. S. Sravnenie vozmozhnostej setej Petri i BAN-logiki v analize kriptograficheskix protokolov proverki podlinnosti i obmena klyuchami / N. S. Mogilevskaya, S. S. Kolchanov // Sistemny`j analiz, upravlenie i obrabotka informacii. — Rostov-na-Donu: Izd. centr DGTU, 2011. — S. 98—101. — In Russian.

7. Aly, S. Protocol verification and analysis using colored Petri nets. Technical report / S. Aly. — Cairo: Cairo University, 2003. — 26 p.

SOFTWARE TOOL FOR LOGICAL VALIDATION OF CRYPTOGRAPHIC KEY GENERATION PROTOCOLS BASED ON BAN-LOGIC

N. S. Mogilevskaya
(Don State Technical University)

The idea of analyzing cryptographic key generation protocols through BAN-logic methods is considered. An example of Kerberos protocol analysis is given. The software tool that automates the analysis of key generation protocols is built. Its validation is investigated.

Keywords: *formal protocol analysis, cryptographic protocols, key generation protocols, computer-aided protocol analysis tools, Kerberos protocol.*