подпись. Можно применить административное управление, обеспечивающее безопасность закрытых ключей, для того чтобы, по крайней мере, хоть в какой-то степени ослабить эти угрозы. Один из возможных способов состоит в требовании в каждую подпись сообщения включать отметку времени (дату и время) и сообщать о скомпрометированных ключах в специальный центр.

Другая угроза состоит в том, что закрытый ключ может быть действительно украден у X в момент времени T. Нарушитель может затем послать сообщение, подписанное подписью X и помеченное временной меткой, которая меньше или равна T.

Описание предлагаемого протокола

Проблемы, связанные с прямой цифровой подписью, могут быть частично решены с помощью арбитра. Для этих целей предлагается следующий криптографический протокол:

(1) Передаваемое сообщение подписывается хэш-ключем симметричной схемой цифровой подписи с модификацией битовых групп [8], используя в качестве блочного шифра алгоритм шифрования «Кузнечик» [1-3, 5, 6, 12, 13].

(2) Устанавливаем виртуальное соединения по протоколу TCP в среде разграничения доступа между арбитрами C и D.

(3) Передаем по каналу связи арбитру C (сервер идентификации отправки) информацию об отправке сообщения получателю B (время отправки, идентификаторы отправителя и получателя). Арбитр пользуется доверием двух сторон (A и B).

(4) Передаем по каналу связи с помощью транспортного протокола TCP зашифрованное сообщение абоненту B.

(5) Получатель отправляет запрос к арбитру C с целью подтвердить отправку сообщения M.

(6) Арбитр C после подтверждения отправки перенаправляет абонента B к арбитру D (сервер проверки подписи), который проверяет цифровую подпись абонента A.

(7) Если подпись верна, арбитр D информирует об этом получателя B.

(8) В случае неверной цифровой подписи, сообщение игнорируется.

Дополнительное разграничение доступа между арбитрами позволяет за счет небольших затрат значительно усложнить задачу осуществления атаки. Под виртуальным соединением в данном случае следует понимать логическую организацию ресурсов IP-сетей, в частности – средств разграничения доступа. Моделью такого средства является монитор безопасности, через который проходят, как запросы от субъекта к объекту, так и ответы объекта, передаваемые субъекту.

Верификация средствами AVISPA

Одним из ключевых этапов при разработке протоколов является стадия проверки на безопасность, которая позволяет проверить их стойкость к различным видам атакам, определить возможность получения защищаемой информации злоумышленником [7].

Таким образом, различные уязвимости будут определяться множеством атак направленных на протоколы. Выделяют следующие основные виды атак [9], представленные на рис. 1.

В источнике [7], справедливо указывается, что не существует универсального средства верификации протоколов, но при этом предлагается ряд ключевых критериев для выбора такого рода средств, один из которых является уровень автоматизации, соответственно, чем он выше, тем более приемлемо средство верификации.

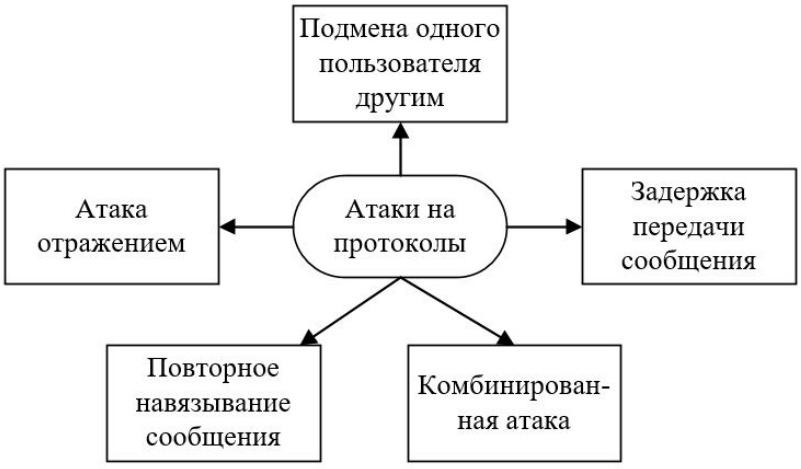

Рис. 1. *Виды атак на протоколы передачи данных*

Выделяют два основных метода автоматизированного анализа основанные на: верификации и логическом выводе [7, 9]. Первый метод базируется на проверке модели, т.е. в формализованном виде получают модель, которая подвергается проверке на соответствие тому или иному критерию безопасности, поэтапно проходя различные собственные состояния. Итогом является выявление путей, приводящих к состоянию угрозы, тем самым непосредственно определяется атака на протокол. Второй метод заключается в применении основных положении логики, а именно, дедукционного анализа на удовлетворение предъявляемых к нему требований.

С точки зрения универсальности процесса верификации протоколов интерес представляет продукт AVISPA, базирующийся на языке описания протоколов HLPLS (High Level Protocol Specification Language). Основным преимуществом данного средства является его многофункциональность - он позволяет не только находить уязвимости у того или иного протокола, но и определять возможные совершаемые атаки на него. Также стоит отметить, что, несмотря на сложность языка HLPLS, язык является достаточно гибким, что дает возможность более детально описывать протоколы по сравнению с аналогами. Результаты, полученные с помощью AVISPA, более подробны и конструктивны. Программа использует язык HLPLS для формализации проверяемых протоколов, но при этом имеется возможность описать сеансы связи при помощи языка CAS+, который имеет более простой синтаксис. При использовании языка CAS+ в дальнейшем можно компилировать CAS+-код в синтаксис языка HLPLS, который затем переводится в более низкоуровневый язык IF, за счет которого и осуществляется верификация [11]. Структурная схема AVISPA приведена на рис. 2.

Как видно из рис. 2, в AVISTA интегрированы четыре основных модуля: «OFMC», «CL-AtSe», «SATMC», «TA4SP». Каждый из модулей представляет собой отчасти уникальный верификатор, который может использоваться как самостоятельно, так и в сочетании с другими модулями.

Модули «OFMC», «CL-AtSe», «SATMC» относятся к первому виду рассмотренных автоматизированных систем анализа протоколов на безопасность, а модуль «TA4SP» базируется на методе логической проверки корректности протокола, в частности, на автоматическом доказательстве.

Описание протокола в синтаксисе языка CAS+

Структура спецификации разработанного протокола, описанного на языке CAS+, состоит из шести основных частей: идентификаторы, сообщение, знания участников протокола, моделирование сессий передачи данных, знания злоумышленника и цели верификации.

Рассмотрим подробно каждую из этих частей.

1) Идентификаторы

Идентификаторы, используемые для описания протокола, могут быть следующих типов: user (имя участника), public_key (открытый ключ), symmetric_key (симметричный ключ), function (функ-

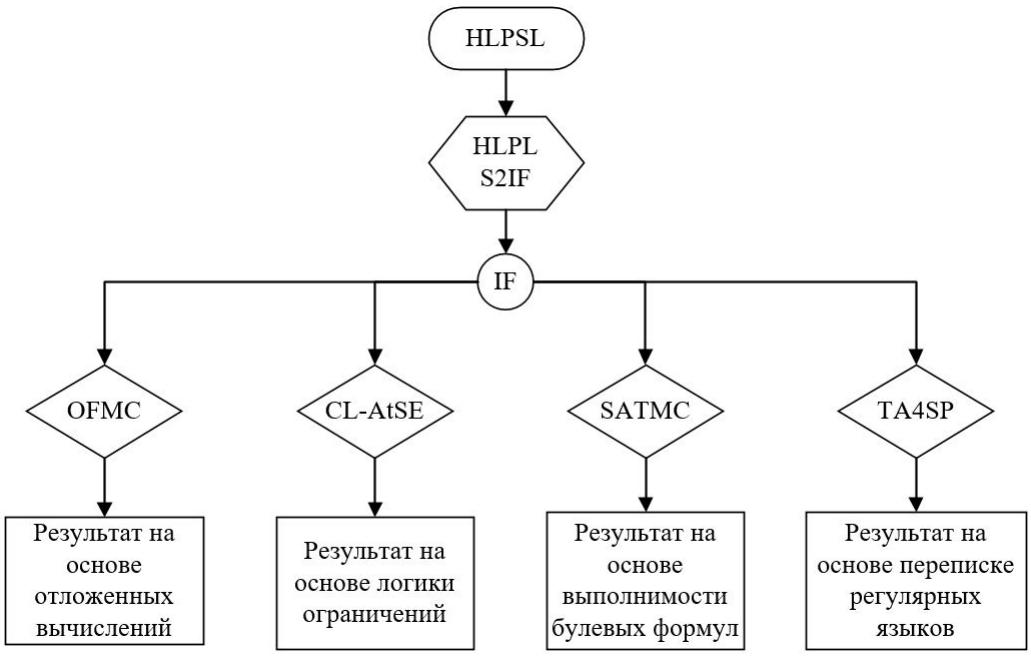

Рис. 2. *Архитектура AVISPA*

ция), number (числовые, текстовые и др. типы данных). К типу function относится в том числе и хэш-функция.

2) Сообщение

Основой для описания правил передачи сообщений между участниками задается следующей формулой:

$$(i.\ S_i \rightarrow_i\ R_i : M_i)\ \ 1 \le i \le n, \qquad (1)$$

где i-номер шага в последовательности передачи сообщений, S_i, R_i - отправитель и получатель соответственно, M_i - сообщение.

Различный тип стрелок определяет тип канала, используемого для передачи. Каналы задаются следующим образом:

- «->» - для использования канала Долева-Яо;
- «=>» - для возможности записи и чтения с защищенных каналов;
- «~>» - для возможности только чтения с защищенных каналов.

3) Знания участников протокола

В поле knowledge задаются знания, которыми владеют участники до начала запуска проверки протокола. Знания им необходимы для инициализации процесса передачи сообщений. При этом предполагается, что каждый участник априорно имеет представление о своем имени.

4) Сессия передачи данных

Эта часть отвечает за присвоение возможных значений постоянных идентификаторов и, таким образом, описываются различные системы, использующие протокол. Здесь также возможно запустить моделирование взаимодействия со злоумышленником, для чего применяется идентификатор *i*. При этом различные ситуационные модели могут происходить как одновременно, так и последовательно произвольное количество раз.

5) Знание злоумышленника

В поле «знание злоумышленника» указывается набор значений, установленных в поле «сессия передача данных» и значения доступные злоумышленнику до начала запуска протокола.

6) Цели верификации

Здесь указываются цели проверки протокола. Существует несколько видов целей: секретность, проверка подлинности различных видов. Секретность связана с соответствующим идентификатором, либо с их набором, которые должны быть указаны в самом начале спецификации. Идентификатор содержит секретные данные, которые должны быть сохранены втайне от других пользователей. Проверка подлинности (аутентификация) связана с двумя пользователями и одним идентификатором, при этом один пользователь производит верификацию другого за счет идентификатора.

Таким образом, описанный выше протокол на языке CAS+ принимает следующий вид (
 - разрыв строки): protocol ecp;
 identifiers
 A, B, C,D: user;
 M, Ok: number;
 K: symmetric_key;
 messages
 1. A -> B : {M}K
 2. A ->C : A
3. C -> B : A, D
 4. B -> D : {M} K
 5. D -> B : Ok <br> knowledge
 A: A,C, B,K;
 B: A,C, B; C: A, B, C, D;
 D: D, K;
 session_instances
 [A:otpr,B:poluch,C:servident, D:servprov,K:key];
 intruder_knowledge
otpr,poluch,servident, servprov.

Цель моделирования, результаты

В данной работе проверка протокола производилась модулем верификации «CL-AtSe», который работает следующим образом: за счет внутриблочного автоматического наложения на протокол ограничений вынуждает злоумышленника выполнять определенные итерации, автоматически контролируя которые, можно идентифицировать атаки на протокол. Каждый этап протокола симулируется ограничениями, становящиеся знаниями злоумышленника, параллельно производится идентификация на соответствие критериям безопасности системы.

Ниже представлено описание протокола на языке HLPLS:

```
// Начало протокола
role role_A(A:agent,C:agent,B:agent,K:symmetric_key,SND,RCV:channel(dy))
played_by A
def=
    local
          State:nat,M:text
    init
          State := 0
    transition
          1. State=0 /\ RCV(start) =|> State'>:=1 /\ M'>:=new() /\ SND({M'>}_K)
/\ SND(A)
end role
```

```
  role role_B(A:agent,C:agent,B:agent,SND,RCV:channel(dy))
  played_by B
  def=
     local
          State:nat,D:agent,K:symmetric_key,M:text,Ok:text
     init
          State := 0
     transition
          1. State=0 /\ RCV({M›}_K›) =|> State›:=1
          3. State=1 /\ RCV(A.D›) =|> State›:=2 /\ SND({M}_K)
          5. State=2 /\ RCV(Ok›) =|> State›:=3
  end role

  role role_C(A:agent,B:agent,C:agent,D:agent,SND,RCV:channel(dy))
  played_by C
  def=
     local
          State:nat
     init
          State := 0
     transition
          2. State=0 /\ RCV(A) =|> State›:=1 /\ SND(A.D)
  end role

  role role_D(D:agent,K:symmetric_key,SND,RCV:channel(dy))
  played_by D
  def=
     local
          State:nat,M:text,Ok:text
     init
          State := 0
     transition
          4. State=0 /\ RCV({M›}_K) =|> State›:=1 /\ Ok›:=new() /\ SND(Ok›)
  end role

  role session1(C:agent,B:agent,A:agent,D:agent,K:symmetric_key)
  def=
     local
          SND4,RCV4,SND3,RCV3,SND2,RCV2,SND1,RCV1:channel(dy)
     composition
          role_D(D,K,SND4,RCV4)     /\     role_C(A,B,C,D,SND3,RCV3)     /\
role_B(A,C,B,SND2,RCV2) /\ role_A(A,C,B,K,SND1,RCV1)
  end role

  role environment()
  def=
     const
          hash_0:function,servprov:agent,poluch:agent,servident:agent,otp
r:agent,key:symmetric_key
     intruder_knowledge = {otpr,poluch,servident,servprov}
     composition
          session1(servident,poluch,otpr,servprov,key)
  end role

  environment()
  // Конец протокола
```

При адекватной компиляции кода, можно получить диаграмму передачи сообщений, которая наглядно описывает работу разработанного протокола (рис. 3).

Затем, запустив модуль верификации «CL-AtSe», в окне программы появляется результат проверки (рис. 4).

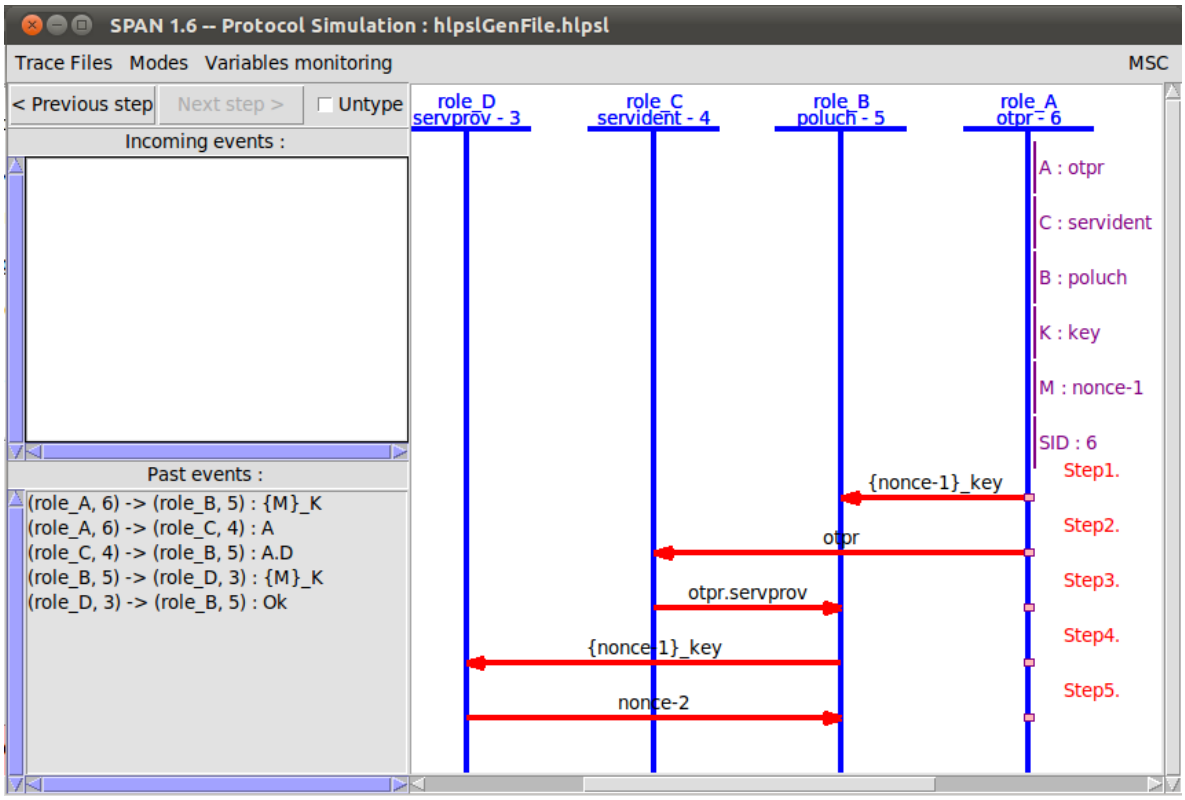

Рис. 3. *Изображение работы протокола в SPAN 1.6:*
A – отправитель, B – получатель, C – сервер идентификации, D – сервер проверки

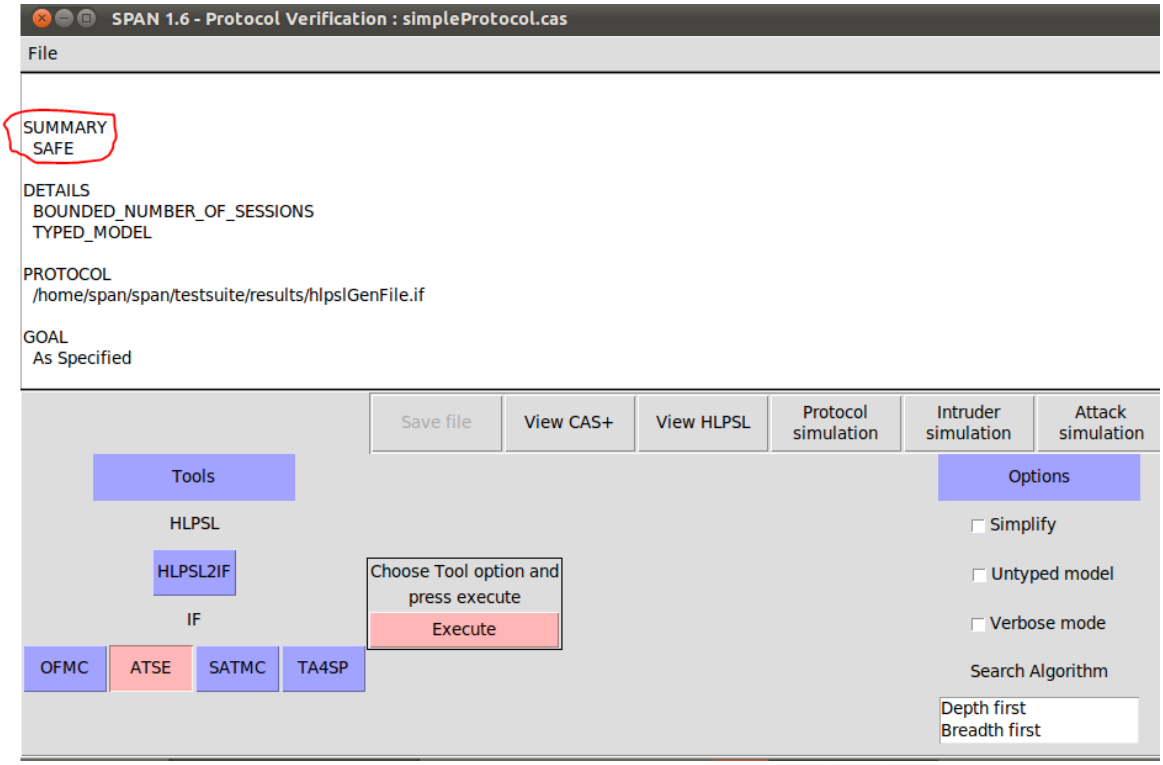

Рис. 4. *Результат проверки протокола модулем «CL-AtSe»*

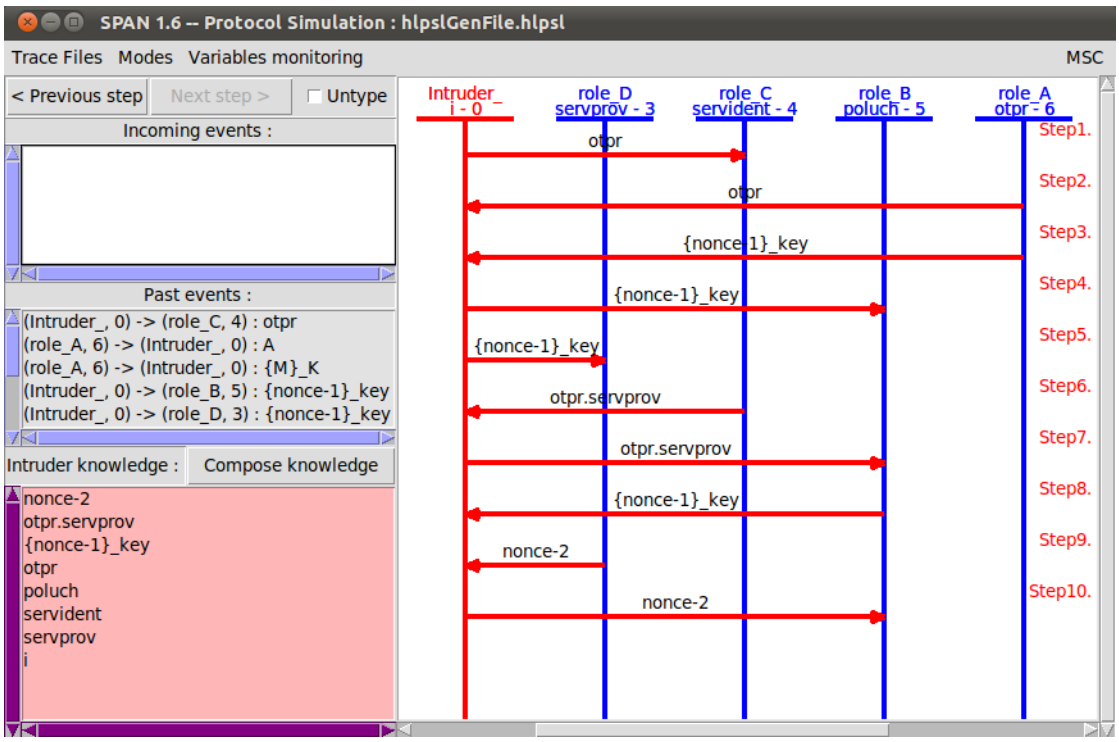

Рис. 5. *Диаграмма передачи сообщений с участием злоумышленника*

Выводы

Таким образом, проверка показала, что протокол оказался защищенным. На рис. 5 представлена диаграмма передачи сообщений при участии злоумышленника. В нижнем левом углу представлены знания злоумышленника, которые он имел до начала сеанса передачи сообщения и после полного его завершения.

Очевидно, что перехватив сообщение *{nonce-1}_key*, ему не заполучить передаваемую информацию без знания ключа шифрования.

Рецензент: *Калмыков Игорь Анатольевич, профессор, доктор технических наук, профессор кафедры информационной безопасности автоматизированных систем Северо – Кавказского федерального университета, email: kia762@yandex.ru*

Литература:

1. Бабенко Л.К. Ищукова Е.А. Сидоров И.Д. Параллельные алгоритмы для решения задач защиты информации. М.: Горячая линия Телеком, 2014. 304 с.
2. Бабенко Л.К., Ищукова Е.А., Ломов И.С. Математическое моделирование криптографического алгоритма «Кузнечик» // Информационное противодействие угрозам терроризма. 2015. № 24. С. 166-176.
3. Бабенко Л.К., Ищукова Е.А., Маро Е.А., Сидоров И.Д., Кравченко П.П. Развитие криптографических методов и средств защиты информации // Известия ЮФУ. Технические науки. 2012. № 4 (129). С. 40-50.
4. Ищукова Е.А., Кошуцкий Р.А., Бабенко Л.К. Разработка и реализация высокоскоростного шифрования данных с использованием алгоритма Кузнечик // Auditorium. 2015. № 4 (08). С. 80-88.
5. Маро Е.А. Реализация алгебраической атаки на шифры ГОСТ Р 34.12-2015 // Инженерный вестник Дона. 2015. Т. 39. № 4-2 (39). С. 4.
6. Резник С.А., Котенко И.В. Методы и средства верификации для комбинированного анализа протоколов безопасности // Защита информации. Инсайд. 2009. № 3 (27). С. 56-72.
7. Санчес Р.Х.А. Разработка цифровой подписи на базе симметричного алгоритма шифрования «Кузнечик» // НАУКА И ИННОВАЦИИ В XXI ВЕКЕ: АКТУАЛЬНЫЕ ВОПРОСЫ, ОТКРЫТИЯ И ДОСТИЖЕНИЯ сборник статей победителей III Международной научно-практической конференции: в 2 частях. 2017. С. 42-45.
8. Чеканов С.Г. Разработка, реализация и анализ криптографического протокола цифровой подписи на основе эллиптических кривых // Вестник Южно-Уральского государственного университета. Серия: Математическое моделирование и программирование. 2013. Т. 6. № 2. С. 120-127.
9. Черемушкин А.В. Автоматизированные средства анализа протоколов // Прикладная дискретная математика. Приложение. 2009. № 1. С. 34-36.
10. Babenko L., Ischukova E., Maro E. GOST Encryption Algorithm and Approaches to its Analysis // Theory and Practice of Cryptography Solutions for Secure Information Systems, IGI Global book series Advances in Information Security, Privacy, and Ethics (AISPE) Book Series, Published in the United States of America by Information Science Reference. 2013. P. 34–61.
11. Babenko L.K., Ishchukova E.A., Maro E.A. Research about Strength of GOST 2814789 Encryption Algorithm // Proceedings of the 5th international conference on Security of information and networks (SIN 2012), ACM, New York, NY, USA. P. 80–84.

VERIFICATION OF THE SECURITY OF THE ELECTRONIC DIGITAL SIGNATURE PROTOCOL USING AVISPA

Babenko L.[3], Jose A. Sánchez[4]

The wide spread of the information networks, particularly, global internet network, and, accordingly, the development of new network services and processing capabilities leads to continuous increase in new security threats to information exchange, which, in its turn, constantly makes the verification procedure and analysis of the data transmission protocols ever more difficult. To ensure security, information networks use interfacing protocols between the elements of these networks. The security is ensured, for instance, by means of confidentiality or authentication, which, in its turn, is implemented by cryptographic systems. One of the key protocol developing stages is a security control stage used to check the protocols for resistance to various types of attacks, define the probability for obtaining sensitive information by a hacker. Nowadays, there is rather a wide range of protocol verification methods for security criteria, however, not all of them, despite meeting the above criteria, can correspond to a high and effective level of verification of major and complex protocols. From this perspective, such protocol test tool as AVISPA, which performs effective verification of a wide spectrum of large-scale protocols, has gained a good reputation. This article provides a protocol of symmetrical digital signature with two attributes in the access control environment, and its verification using AVISPA.

***Keywords**: AVISPA, protocol verification, cryptography, symmetric digital signature, cryptographic protocol*

References:

1. Babenko L.K. Ishchukova E.A. Sidorov I.D. Parallel'nye algoritmy dlya resheniya zadach zashchity informatsii. M.: Goryachaya liniya Telekom, 2014. 304 P.
2. Babenko L.K., Ishchukova E.A., Lomov I.S. Matematicheskoe modelirovanie kriptograficheskogo algoritma «Kuznechik», Informatsionnoe protivodeystvie ugrozam terrorizma. 2015. No 24, pp. 166-176.
3. Babenko L.K., Ishchukova E.A., Maro E.A., Sidorov I.D., Kravchenko P.P. Razvitie kriptograficheskikh metodov i sredstv zashchity informatsii, Izvestiya YuFU. Tekhnicheskie nauki. 2012. No 4 (129), pp. 40-50.
4. Ishchukova E.A., Koshutskiy R.A., Babenko L.K. Razrabotka i realizatsiya vysokoskorostnogo shifrovaniya dannykh s ispol'zovaniem algoritma Kuznechik, Auditorium. 2015. No 4 (08). pp. 80-88.
5. Maro E.A. Realizatsiya algebraicheskoy ataki na shifry GOST R 34.12-2015 // Inzhenernyy vestnik Dona. 2015. T. 39. No 4-2 (39), P. 4.
6. Reznik S.A., Kotenko I.V. Metody i sredstva verifikatsii dlya kombinirovannogo analiza protokolov bezopasnosti, Zashchita informatsii. Insayd. 2009. No 3 (27), pp. 56-72.
7. Sanches R.Kh.A. Razrabotka tsifrovoy podpisi na baze simmetrichnogo algoritma shifrovaniya «Kuznechik», NAUKA I INNOVATsII V XXI VEKE: AKTUAL'NYE VOPROSY, OTKRYTIYa I DOSTIZhENIYa sbornik statey pobediteley III Mezhdunarodnoy nauchno-prakticheskoy konferentsii: v 2 chastyakh. 2017, pp. 42-45.
8. Chekanov S.G. Razrabotka, realizatsiya i analiz kriptograficheskogo protokola tsifrovoy podpisi na osnove ellipticheskikh krivykh, Vestnik Yuzhno-Ural'skogo gosudarstvennogo universiteta. Seriya: Matematicheskoe modelirovanie i programmirovanie. 2013. T. 6. No 2, pp. 120-127.
9. Cheremushkin A.V. Avtomatizirovannye sredstva analiza protokolov, Prikladnaya diskretnaya matematika. Prilozhenie. 2009. no 1. pp. 34-36.
10. Babenko L., Ischukova E., Maro E. GOST Encryption Algorithm and Approaches to its Analysis, Theory and Practice of Cryptography Solutions for Secure Information Systems, IGI Global book series Advances in Information Security, Privacy, and Ethics (AISPE) Book Series, Published in the United States of America by Information Science Reference. 2013, pp. 34–61.
11. Babenko L.K., Ishchukova E.A., Maro E.A. Research about Strength of GOST 2814789 Encryption Algorithm // Proceedings of the 5th international conference on Security of information and networks (SIN 2012), ACM, New York, NY, USA. P. 80–84.

3 Liudmila Babenko, Dr.Sc., professor, Southern Federal University, Taganrog, Russia, email: blk@tsure.r
4 Jose A. Sánchez, Southern Federal University, Caracas, Bolivarian Republic of Venezuela, email: jasroda@gmail.com